

Школа Инженерная школа информационных технологий и робототехники  
 Направление подготовки 09.03.04 Программная инженерия  
 Отделение школы (НОЦ) Отделение информационных технологий

### БАКАЛАВРСКАЯ РАБОТА

Тема работы
<b>Локализация областей кальцинации в лёгочной системе по снимкам компьютерной томографии</b>

УДК 616-073.755:616.24-045.48

Студент

Группа	ФИО	Подпись	Дата
8K51	Пестова Татьяна Витальевна		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Аксёнов Сергей Владимирович	к.т.н.		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСНГ ШБИП	Подопригора Игнат Валерьевич	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Винокурова Галина Фёдоровна	к.т.н.		

### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Евгений Сергеевич	к.т.н.		

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результата	Результат обучения (выпускник должен быть готов)
P1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
P6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
P11	Демонстрировать способность к самостоятельной к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

Министерство науки и высшего образования Российской Федерации  
федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники  
Направление подготовки 09.03.04 Программная инженерия  
Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:  
Руководитель ООП  
\_\_\_\_\_ Чердынцев Е.С.  
(Подпись)      (Дата)      (Ф.И.О.)

### ЗАДАНИЕ на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

Студенту:

Группа	ФИО
8K51	Пестовой Татьяне Витальевне

Тема работы:

<b>Локализация областей кальцинации в лёгочной системе по снимкам компьютерной томографии</b>	
Утверждена приказом директора (дата, номер)	От 18.03.2019 № 11-п/19

Срок сдачи студентом выполненной работы:

19.06.2019

#### ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<b>Исходные данные к работе</b>	Литература с описанием заболеваний лёгочной системы, литература о существующих классификаторах для построения моделей машинного обучения.
---------------------------------	---

<b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>	Изучение исследуемого заболевания и предметной области, аналитический обзор аналогов. Разработка алгоритма для обнаружения областей кальцинации по снимкам компьютерной томографии. Разработка веб-приложения для проведения поиска областей кальцинации по загруженным снимкам компьютерной томографии.
---	--

**Консультанты по разделам выпускной квалификационной работы**

(с указанием разделов)

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Доцент ОСНГ ШБИП Подопригора Игнат Валерьевич
Социальная ответственность	Доцент ООД ШБИП Винокурова Галина Фёдоровна

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	10.09.2018
---	------------

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Аксёнов Сергей Владимирович	к.т.н.		10.09.2018

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8K51	Пестова Татьяна Витальевна		10.09.2018

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа информационных технологий и робототехники  
 Направление подготовки 09.03.04 Программная инженерия  
 Отделение школы (НОЦ) Отделение информационных технологий  
 Период выполнения осенний/весенний семестр 2018/2019 учебного года

Форма представления работы:

Бакалаврская работа
---------------------

### КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	19.06.2019
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
10.10.2018	Обзор предметной области	15
20.02.2019	Разработка модуля обнаружения областей кальцинации	20
20.03.2019	Проектирование веб-приложения для анализа снимков компьютерной томографии на предмет наличия областей кальцинации	20
24.04.2019	Разработка веб-приложения для анализа снимков компьютерной томографии на предмет наличия областей кальцинации	15
02.06.2019	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
02.06.2019	Социальная ответственность	15

**СОСТАВИЛ:**

**Руководитель ВКР**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Аксёнов Сергей Владимирович	к.т.н.		10.09.2018

**СОГЛАСОВАНО:**

**Руководитель ООП**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Евгений Сергеевич	к.т.н.		10.09.2018

# ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8K51	Пестовой Татьяне Витальевне

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

## Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Коэффициент дополнительной заработной платы на стадии проектирования принимается равным 0,12 – 0,15. Коэффициент отчислений во внебюджетные фонды для учреждений, осуществляющих научную и образовательную деятельность, берётся равным 28%. Коэффициент накладных расходов берётся в размере 16% от всех остальных затрат.
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

## Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	QuaD-анализ, SWOT-анализ
2. Планирование и формирование бюджета научных исследований	Планирование этапов работ, определение трудоемкости и построение календарного графика, формирование бюджета
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Оценка сравнительной эффективности исследования

## Перечень графического материала (с точным указанием обязательных чертежей):

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. Альтернативы проведения НИ
4. График проведения и бюджет НИ
5. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику	01.03.2019
--	------------

## Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСНГ ШБИП	Подопригора Игнат Валерьевич	К.Э.Н.		

## Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8K51	Пестова Татьяна Витальевна		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

<b>Группа</b>	<b>ФИО</b>
8K51	Пестовой Татьяне Витальевне

<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение школы (НОЦ)</b>	<b>ОИТ</b>
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

### Локализация областей кальцинации в лёгочной системе по снимкам компьютерной томографии

#### Исходные данные к разделу «Социальная ответственность»:

Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объектом исследования является программная разработка веб-приложения для локализации областей кальцинации в легочной системе по снимкам компьютерной томографии. Данное ПО может использоваться медицинскими учреждениями, которые проводят диагностику заболеваний лёгочной системы, чтобы ускорить процесс постановки диагноза.
---	---

#### Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<b>1. Правовые и организационные вопросы обеспечения безопасности:</b> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	К основным нормативным документам относятся: ГОСТ 12.0.003-2015 ССБТ, ТОИ Р-45-084-01, СанПиН 2.2.4.548–96, СН 2.2.4/2.1.8.562–96, СНиП 23-03-2003, СП 52.13330.2016, СанПиН 2.2.2/2.4.1340-03, ГОСТ Р 50571.17-2000, ГОСТ 33875-2016, ГОСТ 12.3.031-83.
<b>2. Производственная безопасность</b> <ul style="list-style-type: none"> <li>– анализ выявленных вредных и опасных факторов;</li> <li>– анализ выявленных вредных и опасных факторов;</li> </ul>	Вредными факторами труда инженера-программиста являются: недостаточная освещенность рабочей зоны; отклонение параметров микроклимата в помещении; повышенный уровень шума. Опасными факторами труда инженера-программиста являются: опасность поражения электрическим током.
<b>3. Экологическая безопасность:</b> <ul style="list-style-type: none"> <li>– анализ воздействия средств исследования на окружающую среду.</li> </ul>	Утилизация используемой орг. техники и люминесцентных ламп.
<b>4. Безопасность в чрезвычайных ситуациях:</b> <ul style="list-style-type: none"> <li>– обоснование мероприятий по предотвращению наиболее вероятной ЧС и разработка порядка действий в случае возникновения ЧС.</li> </ul>	Возможные ЧС при разработке и эксплуатации проектируемого решения: авария на коммунальных системах жизнеобеспечения, пожар (взрыв) в здании, землетрясение. Наиболее типичной ЧС является пожар (взрыв) в здании.

Дата выдачи задания для раздела по линейному графику	01.03.2019
--	------------

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ООД ШБИП	Винокурова Галина Федоровна	к.т.н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8K51	Пестова Татьяна Витальевна		



## Реферат

Выпускная квалификационная работа содержит 100 страниц, 29 рисунков, 24 таблицы, 4 приложения, 24 использованного источника.

Ключевые слова: кальциноз, компьютерная томография, машинное обучение, Python, веб-приложение, ASP.Net Core.

Объектом исследования являются снимки компьютерной томографии лёгочной системы с установленным диагнозом «Кальциноз».

Цель работы – автоматическое обнаружение областей кальцинации в лёгочной системе по снимкам компьютерной томографии для ускорения процесса постановки диагноза пациентам.

Для реализации поставленной цели необходимо было разработать алгоритм поиска областей кальцинации, а также веб-приложение, позволяющее по загруженным снимкам компьютерной томографии получать обработанные снимки с подсвеченными областями возможных областей кальцинации.

Разработка алгоритма локализации областей кальцинации проводилась с использованием инструментов машинного обучения на языке Python. Веб-приложение было разработано на платформе ASP.Net Core и фреймворке Angular.

## Определения, обозначения и сокращения

КТ – компьютерная томография.

КТ-снимок – снимок, полученный с помощью компьютерного томографа.

Веб-приложение – веб-приложение для анализа КТ-снимков, разрабатываемое в рамках работы.

IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов.

ПО – программное обеспечение.

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

C# – объектно-ориентированный язык программирования, разработанный группой инженеров компании Microsoft.

Java – объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle).

Machine Learning; ML – класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач.

GLCM – сводная таблица состояний, отражающая то, как часто различные комбинации значений яркости пикселя (оттенки серого) встречаются на изображении.

BFS – метод обхода графа и поиска пути в графе, работающий путём последовательного просмотра отдельных уровней графа, начиная с узла-источника.

ЕРС – тип блок-схемы, используемой для бизнес-моделирования в виде событийных цепочек процессов.

DFD – методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

ERD – средство моделирования данных для построения концептуальной схемы базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

БД – база данных.

ASP.Net Core – технология от компании Microsoft, предназначенная для создания различного рода веб-приложений.

MVC (Model-View-Controller) – архитектурный паттерн, разделяющий данные приложения, пользовательский интерфейс и управляющую логику на три отдельных компонента: модель, представление и контроллер – таким образом, что может осуществляться независимая модификация каждого компонента.

Web API – программный интерфейс приложения для веб-сервера или браузера.

Фреймворк – заготовки, шаблоны для программной платформы, определяющие архитектуру программной системы; программное обеспечение, облегчающее разработку и объединение разных модулей программного проекта.

Angular – фреймворк от компании Google для создания клиентских приложений.

Vue.js – JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов.

Microsoft Azure – название облачной платформы Microsoft, предоставляющей возможность разработки и выполнения приложений и хранения данных на серверах, расположенных в распределённых дата-центрах.

Amazon Web Services – коммерческое публичное облако, поддерживаемое и развиваемое компанией Amazon, предоставляющее подписчикам услуги как по инфраструктурной модели (виртуальные серверы, ресурсы хранения), так и платформенного уровня (облачные базы данных, облачное связующее программное обеспечение, облачные бессерверные вычисления, средства разработки).

Google Cloud Platform – предоставляемый компанией Google набор облачных служб, предоставляющий ряд модульных облачных служб, таких как облачные вычисления, хранение данных, анализ данных и машинное обучение.

SPA (Single Page Application) – одностраничное веб-приложение, в котором архитектура устроена таким образом, что при переходе на новую страницу, обновляется только часть контента, и нет необходимости повторно загружать одни и те же элементы.

Страховой номер индивидуального лицевого счёта; СНИЛС – лицевой счёт, содержащийся в страховом свидетельстве обязательного пенсионного страхования, является уникальным для каждого гражданина.

## Оглавление

Введение.....	16
1 Обзор предметной области .....	17
1.1 Моделирование предметной области .....	17
1.2 Обзор аналогов разрабатываемого алгоритма.....	20
1.2.1 Искусственный интеллект в диагностике пневмонии от исследователей Стэнфордского университета.....	20
1.2.2 Алгоритм для анализа флюорографии органов грудной клетки от компании «Google» .....	21
2 Разработка модуля обнаружения областей кальцинации .....	22
2.1 Обоснование выбора языка программирования для реализации модуля.. .....	22
2.2 Описание используемого источника данных.....	23
2.3 Разработка алгоритма обнаружения областей кальцинации с использованием инструментов машинного обучения.....	24
2.3.1 Формирование признаков для построения модели с использованием метода GLCM.....	24
2.3.2 Выбор и тестирование классификатора для построения модели... ..	27
2.4 Разработка алгоритма обнаружения областей кальцинации, основанного на сравнении плотности ткани.....	32
2.4.1 Подготовка данных для алгоритма .....	32
2.4.2 Реализация и тестирование алгоритма .....	35
2.5 Разработка алгоритма окрашивания найденных областей, подозрительных на кальцинацию.....	36
3 Проектирование веб-приложения для анализа снимков компьютерной томографии на предмет наличия областей кальцинации.....	39
3.1 Описание ролей пользователей .....	39
3.2 Проектирование веб-приложения с использованием методологий EPC и DFD .....	39
3.3 Проектирование базы данных веб-приложения с использованием методологии ERD.....	41
3.4 Проектирование диаграммы классов анализа веб-приложения .....	43
3.5 Проектирование макетов страниц веб-приложения.....	44

4	Разработка веб-приложения для анализа снимков компьютерной томографии на предмет наличия областей кальцинации.....	47
4.1	Обоснование выбора средств разработки веб-приложения .....	47
4.2	Взаимодействие веб-приложения с модулем обнаружения областей кальцинации.....	49
4.3	Результат разработки веб-приложения.....	49
5	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение..	52
5.1	Оценка коммерческого потенциала .....	53
5.1.1	Потенциальные потребители результатов исследования .....	53
5.1.2	QuaD-анализ.....	54
5.1.3	SWOT-анализ.....	55
5.2	Определение возможных альтернатив проведения научных исследований .....	57
5.3	Планирование научно-исследовательских работ .....	58
5.3.1	Структура работ в рамках научного исследования .....	58
5.3.2	Определение трудоёмкости выполнения работ .....	60
5.3.3	Бюджет научно-технического исследования (НТИ) .....	62
5.4	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальное и экономической эффективности исследования .....	69
6	Социальная ответственность .....	72
6.1	Правовые и организационные вопросы обеспечения безопасности .....	72
6.1.1	Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства .....	72
6.1.2	Организационные мероприятия при компоновке рабочей зоны	74
6.2	Профессиональная социальная безопасность.....	75
6.2.1	Анализ вредных факторов, которые могут возникнуть при проведении исследований .....	76
6.2.2	Анализ опасных факторов, которые могут возникнуть при проведении исследований .....	80
6.3	Экологическая безопасность .....	81
6.3.1	Анализ воздействия средств исследования на окружающую среду .....	82
6.4	Безопасность в чрезвычайных ситуациях .....	83

6.4.1 Обоснование мероприятий по предотвращению наиболее типичной ЧС и разработка порядка действий в случае возникновения ЧС .....	83
Заключение .....	86
Список использованных источников .....	87
Приложение А .....	90
Приложение Б .....	92
Приложение В.....	95
Приложение Г .....	97

## **Введение**

В настоящее время во многие области медицины проникают информационные технологии. Автоматизированные системы позволяют более точно и быстро устанавливать наличие того или иного заболевания.

Одной из медицинских задач, информационное решение для которой требует разработки, является автоматизированный поиск участков кальциноза в легочной системе по снимкам компьютерной томографии.

Кальциноз – это заболевание, при котором в тканях образуются кальцинаты, являющиеся образованиями, которые формируются в результате скопления солей кальция. По плотности кальцинаты сопоставимы с костной тканью [1]. Одним из способов диагностирования кальциноза является проведение КТ.

Автоматизированный анализ снимков необходим, так как ручной просмотр КТ-снимков влечет за собой проблему длительной обработки, что замедляет постановку диагноза.

Цель разрабатываемой системы – достигнуть быстрой постановки диагноза.

Для достижения этой цели были поставлены следующие задачи:

1. изучить предметную область (процесс приёма пациентов в поликлинике);
2. разработать алгоритм, определяющий предполагаемый диагноз с точностью не менее 90% (области кальцинации обнаружены либо не обнаружены);
3. спроектировать и разработать веб-приложение, которое после загрузки пользователем КТ-снимков анализирует их, выполняя поиск подозрительных на кальциноз участков, окрашивает предполагаемые участки и выдает уже обработанные снимки.



# 1 Обзор предметной области

## 1.1 Моделирование предметной области

Предметной областью разрабатываемой системы является диагностика заболеваний легочной системы. Этот процесс охватывает такие подпроцессы как:

1. регистрация клиентов и управление их профилями;
2. приём у врача-диагностика (далее Врач);
3. приём у лечащего врача.

Для описания исследуемых процессов был составлен ряд диаграмм по методологии IDEF0.

Контекстная диаграмма IDEF0 (рисунок 1) отражает общий процесс в виде «черного ящика»: для процесса указаны входы ( $I1$ ,  $I2$ ) – объекты, используемые и преобразуемые работой для получения результата; выходы ( $O1$ ,  $O2$ ) – объекты, в которые преобразуются входы; управление ( $C1$ ,  $C2$ ,  $C3$ ,  $C4$ ) – информация, управляющая действиями работы; механизмы ( $M1$ ,  $M2$ ,  $M3$ ,  $M4$ ,  $M5$ ) – ресурсы, выполняющие работу [2].

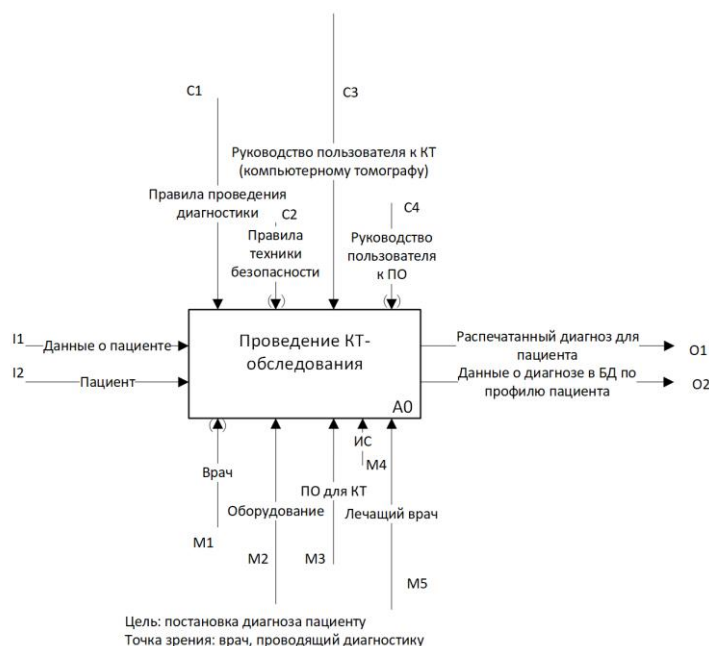


Рисунок 1. Контекстная диаграмма IDEF0

Декомпозиция контекстной диаграммы IDEF0 представлена на рисунке 2.

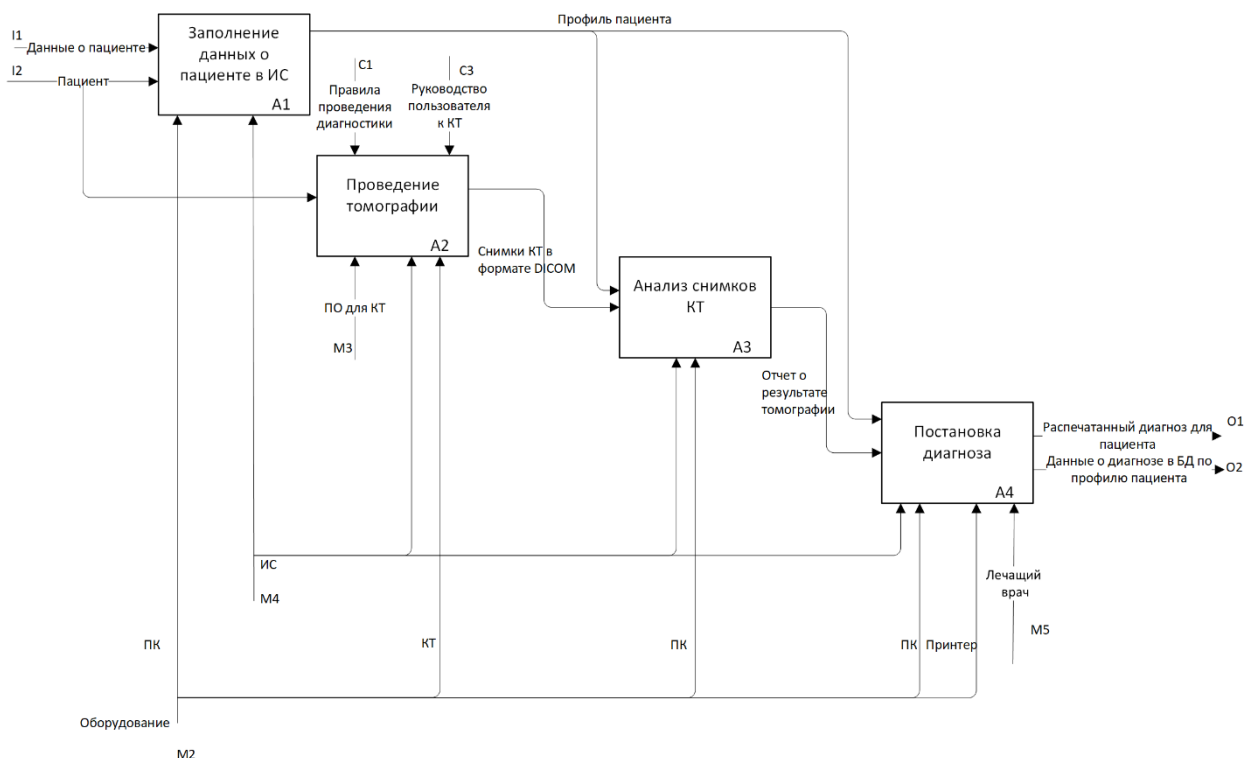


Рисунок 2. Декомпозиция 1-го уровня IDEF0

Декомпозиция 1-го уровня состоит из 4-х процессов: «Заполнение данных о пациенте» (A1), «Проведение томографии» (A2), «Анализ снимков КТ» (A3), «Постановка диагноза» (A4). Проведение томографии проводится с использованием компьютерного томографа и специализированного ПО.

Процесс «Анализ снимков КТ» (A3) – это составляющая общего процесса, которая замедляет процесс постановки диагноза. На диаграмме Исикавы (рисунок 3) представлены источники проблемы длительного анализа КТ-снимков.

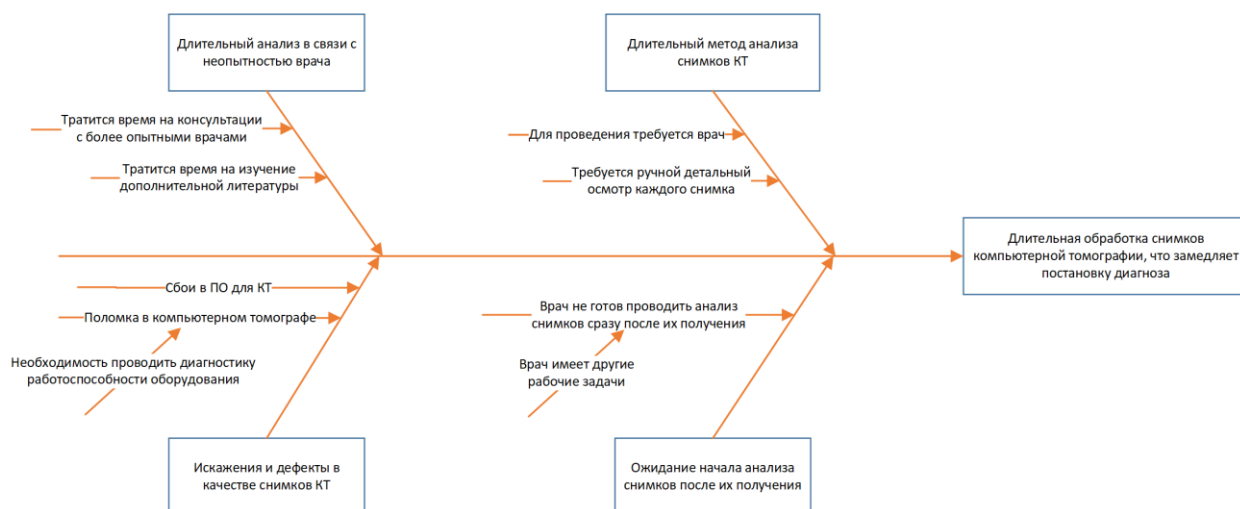


Рисунок 3. Диаграмма Исикавы

Из диаграммы Исикавы следует, что «Анализ снимков КТ» необходимо автоматизировать. Для лучшего понимания этого процесса для него были составлены декомпозиции 2-го уровня «до» и «после» внедрения разработки (рисунки 4-5).

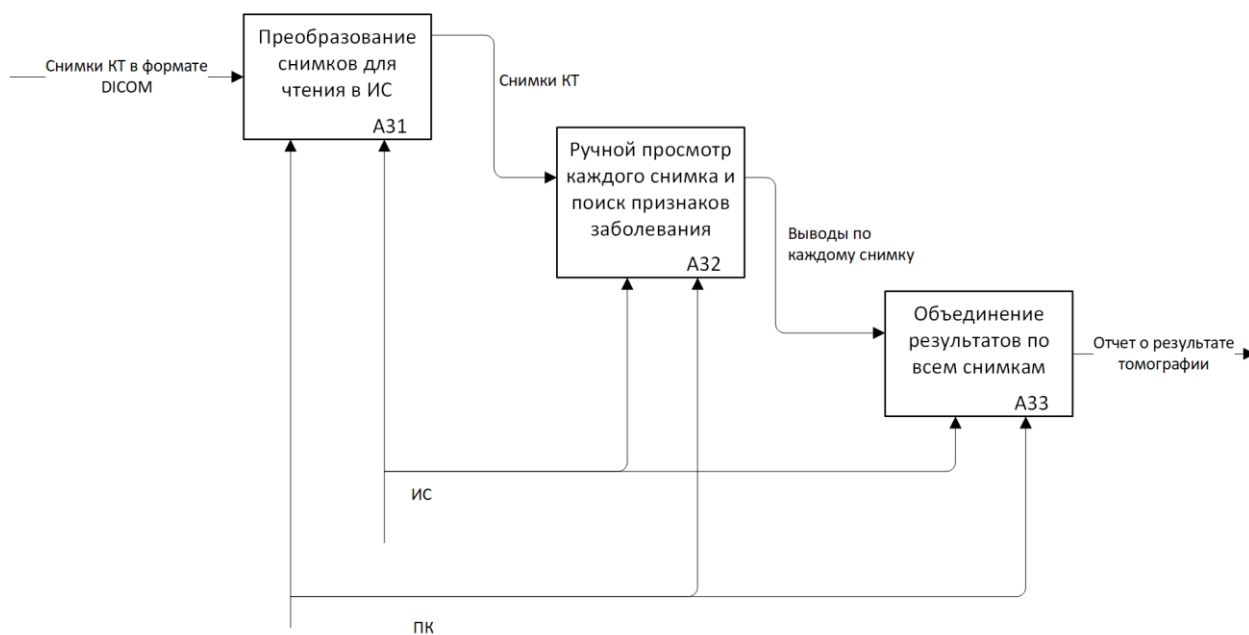


Рисунок 4. Декомпозиция 2-го уровня – «до»

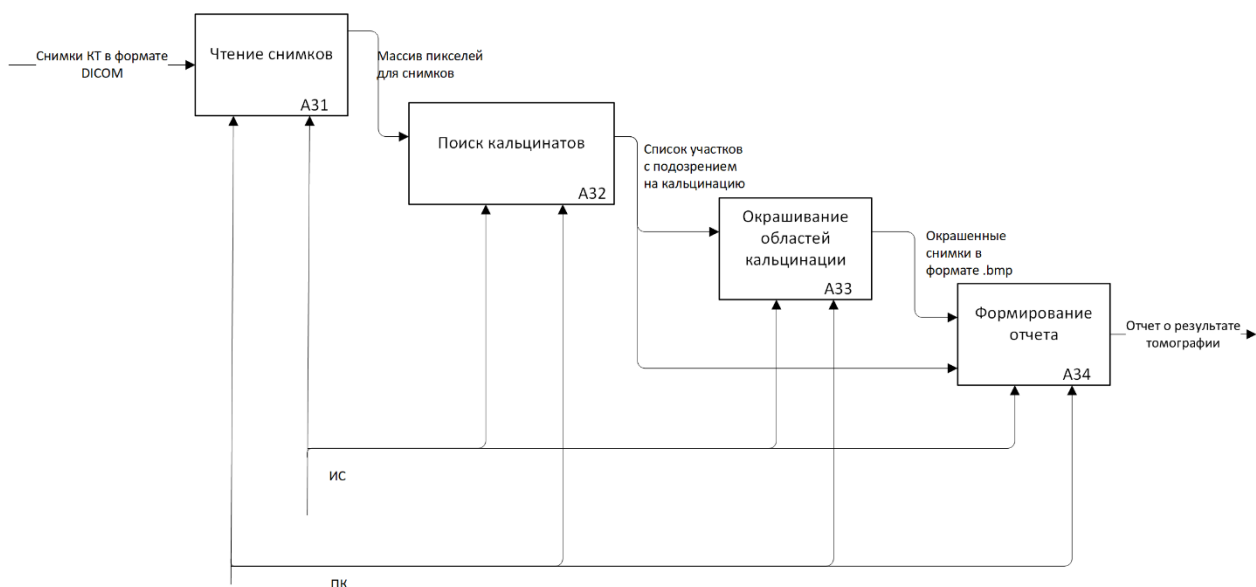


Рисунок 5. Декомпозиция 2-го уровня– «после»

## 1.2 Обзор аналогов разрабатываемого алгоритма

Из-за специфики разрабатываемого алгоритма (поиск участков кальцинации в лёгочной системе) полных аналогов найдено не было. Тем не менее, были найдены примеры разработок со схожими задачами. Однако, все найденные примеры находятся на стадии разработки. Переход к внедрению разработок, связанных с машинным обучением, для постановки диагнозов по медицинским снимкам — длительный процесс из-за необходимости проведения многочисленных испытаний перед внедрением.

### 1.2.1 Искусственный интеллект в диагностике пневмонии от исследователей Стэнфордского университета

В ноябре 2017 года исследователи из Стэнфордского университета представили самообучающийся алгоритм «CheXNet», который способен ставить диагноз пневмонии по рентгенограммам легких [3].

Разработчики научили систему пометить разными цветами те отделы легких, где машина «увидела» признаки воспаления легких: чем ярче цвет — тем вероятнее патология. После машинной обработки рентгенограммы просматривает врач, обращая внимание в первую очередь на те области, которые машина пометила как самые «горячие».

### **1.2.2 Алгоритм для анализа флюорографии органов грудной клетки от компании «Google»**

В конце марта 2018 года компания «Google» продемонстрировала искусственный интеллект для быстрой и эффективной обработки флюорографии органов грудной клетки [3].

Разработчики создали модель глубинного обучения, способную совершенствоваться на основе небольшого числа аннотированных (с помеченными вручную областями, на которых видны отклонения) медицинских снимков и позволяющую одновременно идентифицировать болезнь и выделить ее на изображении.

## 2 Разработка модуля обнаружения областей кальцинации

### 2.1 Обоснование выбора языка программирования для реализации модуля

Перед началом разработки алгоритма необходимо было определиться, какой язык программирования использовать. Для этого была составлена матрица морфологического анализа для возможных вариантов (таблица 1).

Таблица 1 – Выбор языка программирования для реализации алгоритма [4]

Критерий	Весовой коэффициент критерия	Вариант		
		Python	C#	Java
Опыт работы	0,2	5	5	4
Уровень готовых библиотек по ML и их поддержка	0,25	5	3	4
Уровень готовых библиотек для обработки изображений и их поддержка	0,25	5	4	4
Скорость работы	0,2	3	4	4
Потребность в ресурсах памяти	0,1	4	4	3
ИТОГО	1	4,5	3,95	3,9

Таким образом, в качестве языка программирования для разработки алгоритма обнаружения областей кальцинации был выбран Python.

## 2.2 Описание используемого источника данных

В качестве источника данных использовался Belarus tuberculosis portal [5], на котором расположены КТ-снимки реальных пациентов с указанием диагноза. Для анализа были взяты данные четырёх пациентов с исследуемым диагнозом.

При проведении компьютерной томографии лёгочной системы на выходе получают набор снимков, который состоит из около ста (100) снимков в монохромном представлении. Анализировать необходимо не только каждый отдельный снимок, но и изменения между разными снимками («слоями лёгочной системы»). Другими словами, анализ должен проводиться не в 2D-, а в 3D-режиме.

На рисунке 6 представлен пример одного из снимков, используемого для разработки и тестирования алгоритма. Красным выделены области кальцинации.

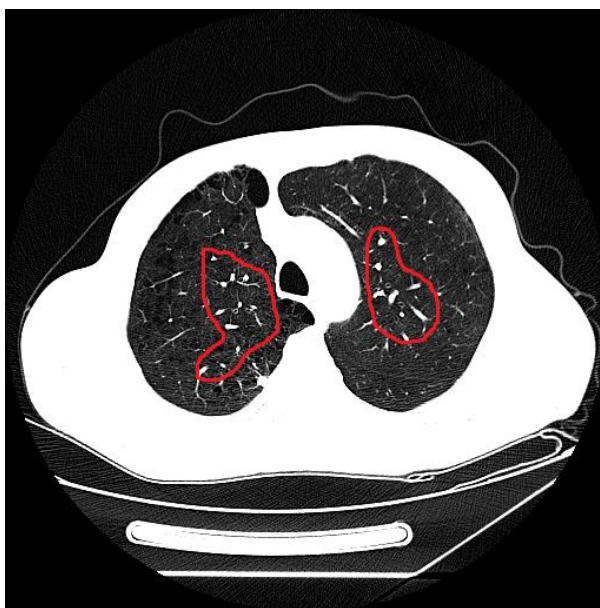


Рисунок 6. Пример КТ-снимка лёгочной системы

## **2.3 Разработка алгоритма обнаружения областей кальцинации с использованием инструментов машинного обучения**

### **2.3.1 Формирование признаков для построения модели с использованием метода GLCM**

Для получения признаков для построения модели использовалась матрица взаимной встречаемости (GLCM – Gray-Level Co-occurrence Matrix).

Для получения значений признаков размеченная часть снимков делилась на параллелепипеды, размером  $x = 10$ ,  $y = 10$ ,  $z = 3$ . Под координатой  $z$  подразумевается «слой» легочной системы.

Затем по каждому участку формировались два признака с помощью метода GLCM: «различие» («dissimilarity») и «соотношение» (correlation). Данные признаки вычислялись с помощью сторонних библиотек Python: «greyscomatrix», «greyscoprops» [6].

Также для получения еще 6-ти признаков использовался собственный алгоритм, схожий на GLCM, который формировал дополнительные признаки:

1. средний цвет всего параллелепипеда;
2. средний цвет восьми подпараллелепипедов, полученных из параллелепипеда разделением его на две части по каждой из сторон;
3. средний цвет каждого «слоя» параллелепипеда;
4. три суммы количества подпараллелепипедов определенного среднего цвета, расположенных на расстоянии 1 от другого подпараллелепипеда (внешние подпараллелепипеды не рассматриваются).

Под суммой количества подпараллелепипедов определенного среднего цвета понимается следующее:



1. разделим весь ранг цветов на черный(1) – черносерый(2) – черносеросерый(3) – серый(4) – серосеробелый(5) – серобелый(6) – белый(7)

2. для каждого внутреннего подпараллелепипеда посчитаем количество подпараллелепипедов, на расстоянии 1, средний цвет которых лежит в промежутке от 2 до 3, в промежутке от 3 до 5, в промежутке от 5 до 7.

Для сокращения расчетов среднего цвета подпараллелепипедов использовалась структура хранения данных – частичные суммы в трехмерном пространстве.

Для объяснения этой структуры данных сначала рассмотрим одномерные частичные суммы. Пусть имеется массив  $A$ , массив  $S$  будет массивом частичных сумм. Тогда верны следующие формулы 1-3.

$$S_0 = A_0, \quad (1)$$

где  $S_0$  – нулевой элемент массива частичных сумм;

$A_0$  – нулевой элемент исходного массива.

$$\forall i_1^{n-1} S_i = S_{i-1} + A_i, \quad (2)$$

где  $i$  – индекс элемента;

$n$  – количество элементов в массиве;

$S$  – массив частичных сумм;

$A$  – исходный массив.

$$\sum_{i_1}^{i_2} A = S_{i_2} - S_{i_1-1}, \quad (3)$$

где  $i_1$  – индекс элемента, начиная с которого необходимо посчитать сумму;

$i_2$  – индекс элемента, заканчивая которым необходимо посчитать сумму;

$A$  – исходный массив;

$S$  – массив частичных сумм.

В случае трехмерных частичных сумм имеем (формула 4):

$$S_{i_1, j_1, k_1} = \sum_{i=0}^{i_1} \sum_{j=0}^{j_1} \sum_{k=0}^{k_1} A_{i, j, k}, \quad (4)$$

где  $i_1$  – индекс элемента по  $x$ -координате, для которого вычисляется значение частичной суммы;

$j_1$  – индекс элемента по  $y$ -координате, для которого вычисляется значение частичной суммы;

$k_1$  – индекс элемента по  $z$ -координате, для которого вычисляется значение частичной суммы;

$S$  – массив частичных сумм;

$A$  – исходный массив.

Формирование трехмерных частичных сумм аналогично формированию одномерных (формула 5):

$$\sum_{i_1, j_1, k_1}^{i_2, j_2, k_2} A = \begin{aligned} & S_{i_2, j_2, k_2} - S_{i_2, j_2, k_1} - S_{i_2, j_1, k_2} - S_{i_1, j_2, k_2} + S_{i_1, j_2, k_1} + \\ & + S_{i_2, j_1, k_1} + S_{i_1, j_1, k_2} - S_{i_1, j_1, k_1} + A_{i_1, j_1, k_1} + \sum_{i=x_1}^{x_2+1} \sum_{j=y_1}^{y_2+1} A_{i, j, k_1} + \\ & + \sum_{j=y_1}^{y_2+1} \sum_{k=k_1}^{k_2+1} A_{i_1, j, k} + \sum_{i=x_1}^{x_2+1} \sum_{k=k_1}^{k_2+1} A_{i, j_1, k} - \\ & - \sum_{i=x_1}^{x_2+1} A_{i, j_1, k_1} - \sum_{j=j_1}^{j_2+1} A_{i_1, j, k_1} - \sum_{k=k_1}^{k_2+1} A_{i_1, j_1, k} \end{aligned}, \quad (5)$$

где  $i_1$  – индекс элемента по  $x$ -координате, начиная с которого необходимо посчитать сумму;

$j_1$  – индекс элемента по  $y$ -координате, начиная с которого необходимо посчитать сумму;

$k_1$  – индекс элемента по  $z$ -координате, начиная с которого необходимо посчитать сумму;

$i_2$  – индекс элемента по  $x$ -координате, заканчивая которым необходимо посчитать сумму;

$j_2$  – индекс элемента по  $y$ -координате, заканчивая которым необходимо посчитать сумму;

$k_2$  – индекс элемента по  $z$ -координате, заканчивая которым необходимо посчитать сумму;

$A$  – исходный массив;

$S$  – массив частичных сумм.

Таким образом, формировалось 8 признаков для дальнейшего построения модели машинного обучения.

### **2.3.2 Выбор и тестирование классификатора для построения модели**

Для задачи классификации рассмотрены следующие алгоритмы:  $k$ -ближайших соседей, дерево решений, случайный лес, градиентный бустинг деревьев решений.

#### **2.3.2.1 Классификатор « $K$ -ближайших соседей»**

Классификатор « $K$ -ближайших соседей» формирует прогноз для новой точки данных, основываясь на типе  $k$ -ближайших к ней точек обучающего набора [7]. Изменяемый параметр этого алгоритма – количество соседей, по которым определяется тип нового элемента.

На рисунке 7 представлена точность работы классификатора при различном количестве соседей.

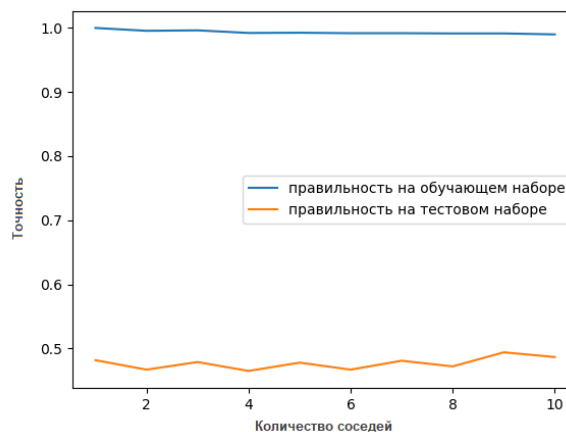


Рисунок 7. Точность работы классификатора « $K$ -ближайших соседей» при различном количестве соседей

Как видно из рисунка 7, лучшая точность классификатора достигается при количестве соседей, равное 9. Тем не менее, правильность на тестовом наборе слишком низка, чтобы использовать этот классификатор для реальной задачи.

#### 2.3.2.2 Классификатор «Дерево решений»

Деревья решений представляют собой иерархию правил «если ..., то», по которым и определяется классифицируется новая точка. Изменяемый параметр этого алгоритма — максимальная глубина дерева, который позволяет избежать переобучения модели [7].

На рисунке 8 представлена точность работы алгоритма при различной максимальной глубине дерева.

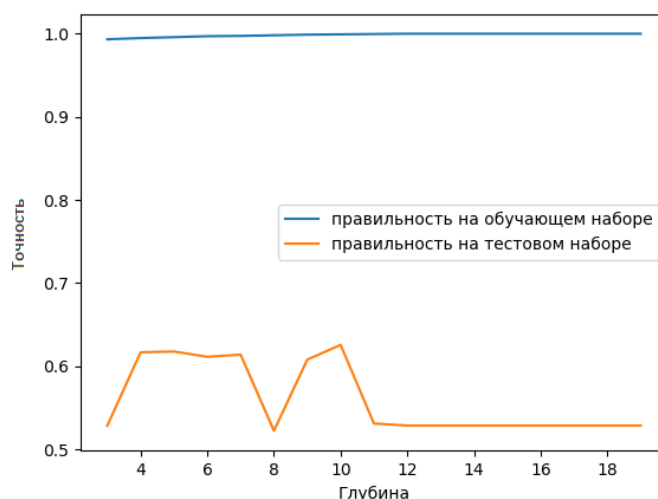


Рисунок 8. Точность работы классификатора «Дерево решений» при различной максимальной глубине дерева

Из рисунка 8 видно, что максимальная точность классификатора достигается при максимальной глубине дерева равной 10.

### 2.3.2.3 Классификатор «Случайный лес»

Случайный лес – это набор деревьев решений, где каждое дерево немного отличается от остальных. Построение нескольких деревьев позволяет достигнуть меньшей степени переобучения, так как каждое дерево обучается индивидуально, а конечный результат получается путем усреднения результатов, полученных от каждого дерева. Один из изменяемых параметров этого классификатора – максимальное количество отбираемых признаков для каждого дерева. Отбор подмножества признаков повторяется отдельно для каждого узла, поэтому в каждом узле дерева может быть принято решение с использованием «своего» подмножества признаков [7].

На рисунке 9 представлена точность работы алгоритма при различном количестве отбираемых признаков.

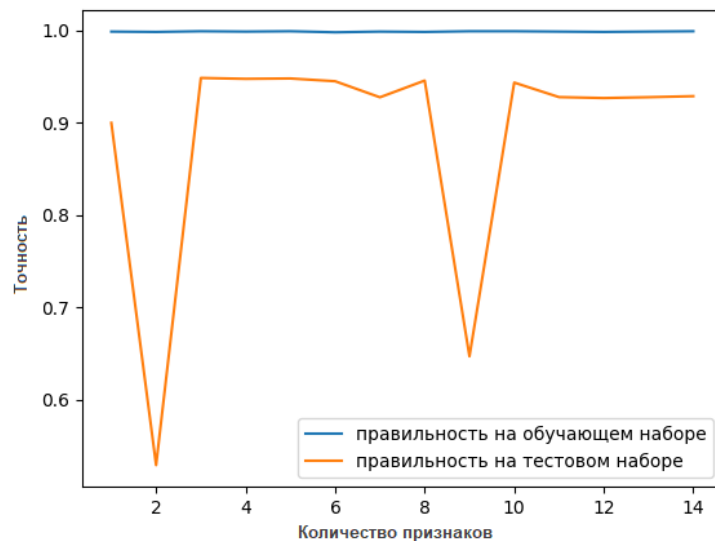


Рисунок 9. Точность работы классификатора «Случайный лес» при различном количестве отбираемых признаков

На рисунке 9 видно, что лучший результат работы классификатора достигается, когда максимальное количество отбираемых признаков равно 8.

#### 2.3.2.4 Классификатор «Градиентный бустинг деревьев решений»

Основная идея градиентного бустинга деревьев решений заключается в объединении множества простых моделей деревьев решений небольшой глубины. Каждое дерево может дать хорошие прогнозы только для части данных и таким образом для итеративного улучшения качества добавляется все большее количество деревьев. Изменяемые параметры этого алгоритма – максимальная глубина дерева и коэффициент обучения – насколько сильно каждое дерево будет пытаться исправить ошибки предыдущих деревьев [7].

На рисунке 10 представлена точность работы алгоритма при различных максимальной глубине деревьев и коэффициенте обучения.

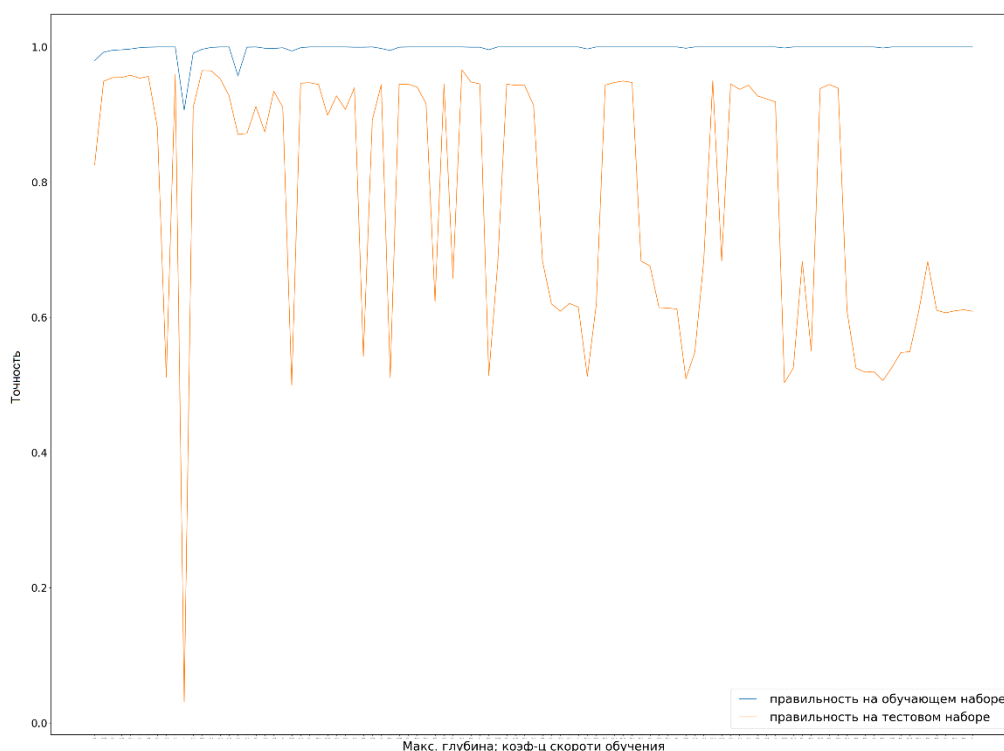


Рисунок 10. Точность работы классификатора «Градиентный бустинг деревьев решений» при различных максимальной глубине деревьев и коэффициенте обучения

Таким образом, лучший результат работы классификатора достигается при максимальной глубине дерева, равной 4, и коэффициенте обучения, равным 0,8.

### 2.3.2.5 Сравнение протестированных классификаторов

Для сравнения точности работы протестированных классификаторов была построена сводная диаграмма этих классификаторов с параметрами, при которых они показывали наилучший результат (рисунок 11, листинг А.1 приложения А).

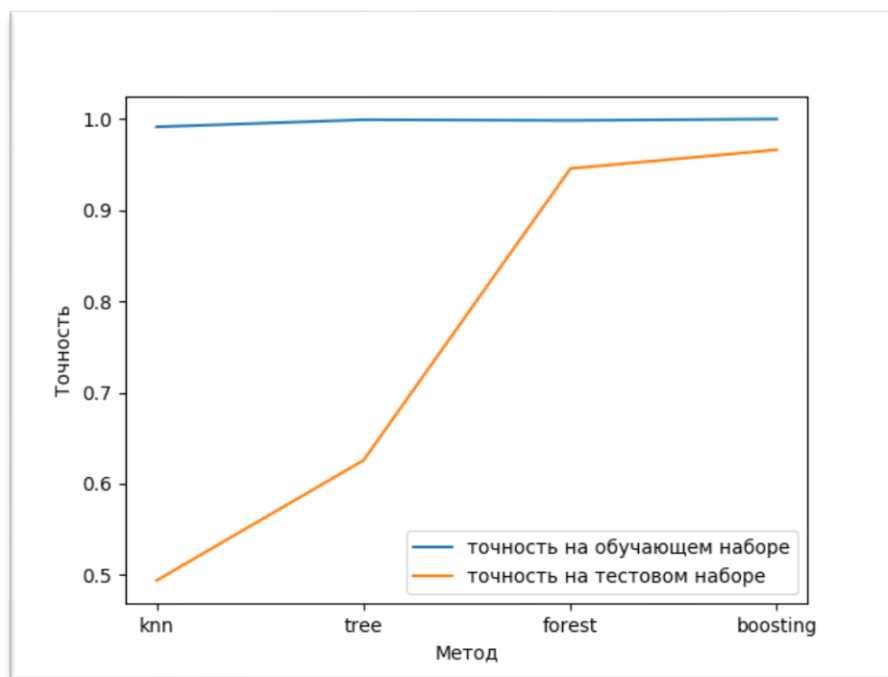


Рисунок 11. Сравнение рассмотренных классификаторов

Из рисунка 11 видно, что на обучающем наборе все классификаторы показывают хороший результат, однако на тестовом наборе наивысшую точность показывает классификатор «Градиентный бустинг деревьев решений». Этот классификатор и был выбран для построения модели и дальнейшего использования в разрабатываемом модуле.

## **2.4 Разработка алгоритма обнаружения областей кальцинации, основанного на сравнении плотности ткани**

Характерной чертой исследуемого заболевания является то, что плотности ткани в зараженных участках очень высокая по сравнению с другими заболеваниями. Исходя из этого была выдвинута гипотеза, что поиск зараженных участков можно проводить, основываясь только на одном признаке – плотности ткани.

### **2.4.1 Подготовка данных для алгоритма**

Снимок компьютерной томографии лёгочной системы содержит не только область легких, поэтому необходимо выделить эту область, чтобы анализ проводился только в нужной части. Для этого использовался уже



реализованный модуль, однако этот модуль давал недостаточно точную сегментацию области лёгких. Он выделял не только область внутри легких, но также и захватывал часть области кости. На рисунке 12 представлен пример работы этого алгоритма (зеленым подсвечена область, которая была определена как область легких и действительно является ей, а синим – которая определена областью легких, но не является ей).

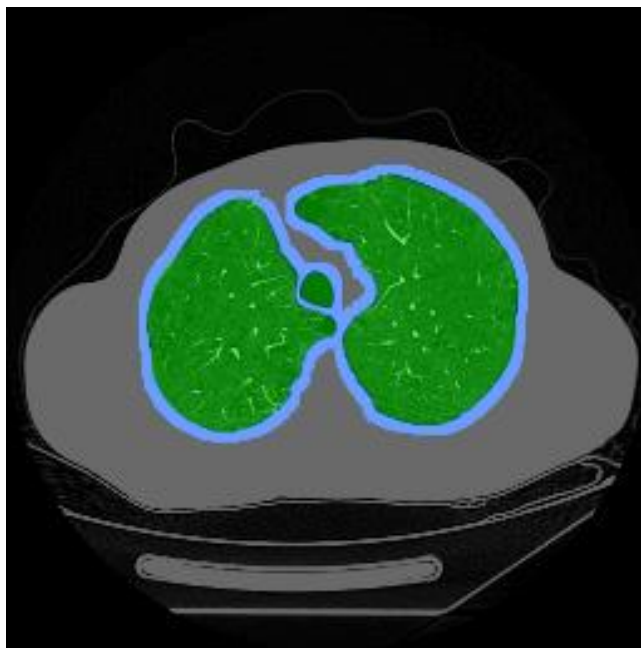


Рисунок 12. Область, определенная как область внутри лёгких

Для некоторых задач такое выделение не является критичным, однако в данном случае это приведет к ошибкам. Область кости является такой же плотной, как и кальцинаты, поэтому при анализе по плотности в область кальцинатов будут попадать участки, которые являются обычными костями.

Для решения этой проблемы был разработан следующий алгоритм, использующий поиск в ширину (BFS) [8]:

1. добавляем в очередь все пиксели, которые определены как «не область легких» и при этом хотя бы один соседний (справа, снизу, слева, сверху) определен как область легких;

2. запускаем поиск в ширину по 4-м направлениям (вправо, вниз, влево, вверх), добавляя в очередь и отмечая как «не область легких» пиксели, которые удовлетворяют следующим условиям:

2.1. их еще не посещали;

2.2. они определены как область легких;

2.3. плотность в этих пикселях больше значения так называемого «цвета останова» – значения плотности, которое соответствует области внутри легких без патологий.

Результат этого алгоритма представлен на рисунке 13.

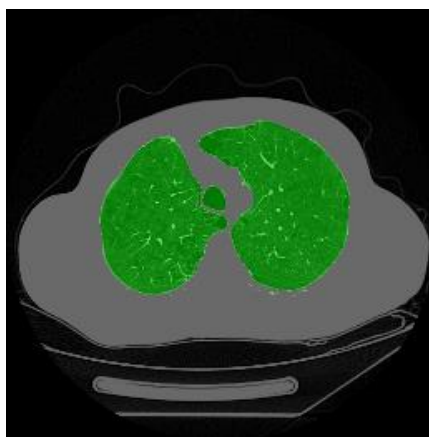


Рисунок 13. Результат работы модифицированного алгоритма сегментации области легких

Как видно из рисунка 13, алгоритм выделяет область легких не гладко, а «зубчиками». Для сглаживания области был использован метод размывания (erosion) [9], результат которого представлен на рисунке 14.

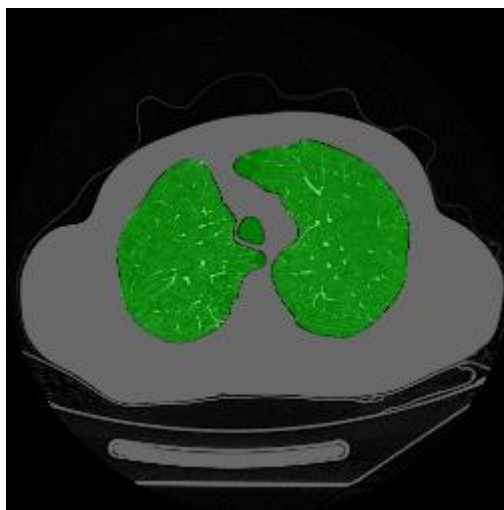


Рисунок 14. Результат работы модифицированного алгоритма сегментации области легких с применением размывания

#### 2.4.2 Реализация и тестирование алгоритма

После того, как область легких была определена, стало возможным приступать к определению места расположения кальцинатов. Алгоритм их определения заключается в том, чтобы проверить, лежит ли значение плотности выше границы «цвета кальцинатов» – значение, при котором пиксель является частью области кальцинации. Эта граница была определена экспериментально и принимает значение 900.

Результат работы алгоритма представлен на рисунке 15 – зеленым выделены участки кальцинации.

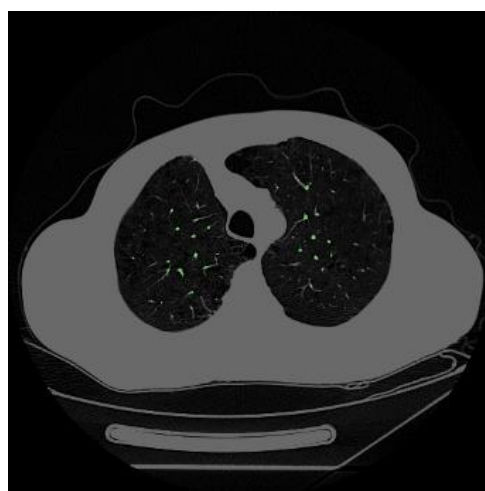


Рисунок 15. Результат работы алгоритма поиска кальцинатов

Тестирование показало, что алгоритм работает с довольно высокой точностью – 87,5%. Тем не менее, алгоритм может начать давать неверный результат при наличии других заболеваний, при которых появляются участки с такой же высокой плотностью ткани. Кроме того, могут возникнуть проблемы, если лёгкие сегментированы не на две доли, а на три и более. Исходя из этого, было решено использовать в дальнейшем алгоритм, основанный на машинном обучении.

## **2.5 Разработка алгоритма окрашивания найденных областей, подозрительных на кальцинацию**

Окрашивание подозрительных на кальцинацию участков снимков производится путем накладки полупрозрачного слоя на исходный снимок. Полупрозрачный слой имеет окраску в участках, предсказанных моделью как зараженные. В результате получается изображение, на котором видны не только области зараженных участков, но и то, что находится внутри этого участка.

На вход модулю окраски подается список, содержащий номер снимка, координаты зараженных участков (пикселей). Также на вход подается путь к папке, в которой находятся оригинальные снимки в формате *.bmp*, и путь для сохранения результирующих изображений.

В листинге 1 представлен код для модуля окраски снимков.

### **Листинг 1. Модуль окраски снимков**

```
from scipy.misc.pilutil import Image
import os
from shutil import copyfile

def color_image_from_coordinate_list(image_path_from,
                                     image_path_to,
                                     coordinate_list):
    im_origin = Image.open(image_path_from)
    im = Image.new('RGB', im_origin.size)
    pix = im.load()
    coordinates_number = len(coordinate_list)
    # color image
```

```

for i in range(0, coordinates_number):
    x = int(coordinate_list[i][0])
    y = int(coordinate_list[i][1])
    pix[x,y] = (0,200,0)
# add transparency to an image
im.putalpha(150)
# combine original image and transparent image
im_origin.paste(im.convert('RGB'), (0,0), im)
im_origin.save(image_path_to)
im_origin.close()
im.close()

def color_images_from_coordinate_list(images_path_from,
                                     images_path_to,
                                     coordinate_list):
    images = os.listdir(images_path_from)
    images_number = len(images)
    for i in range(0, images_number):
        if(len(coordinate_list[i]) == 0):
            copyfile(images_path_from + '/' + images[i],
images_path_to + '/colored_' + images[i])
            continue
        color_image_from_coordinate_list(images_path_from + '/' +
images[i],
images_path_to + '/colored_' + images[i], coordinate_list[i])

```

Демонстрация работы модуля окрашивания представлена на рисунках 16-17.

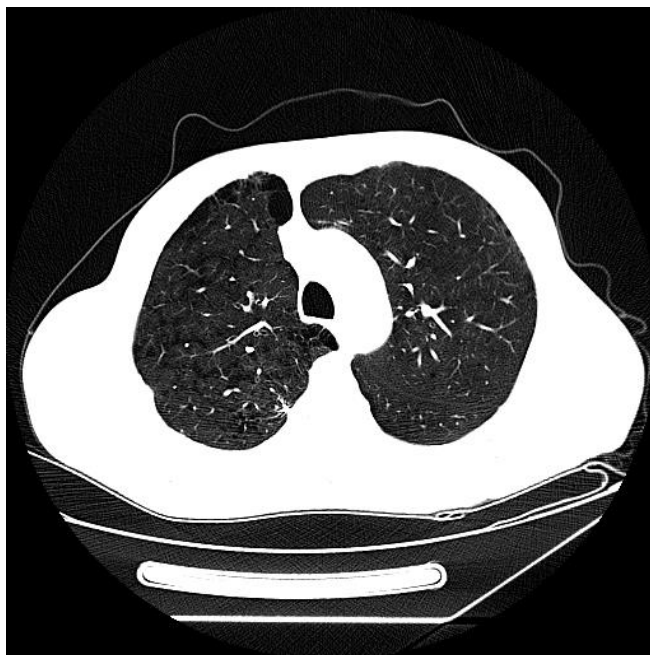


Рисунок 16. Исходное изображение

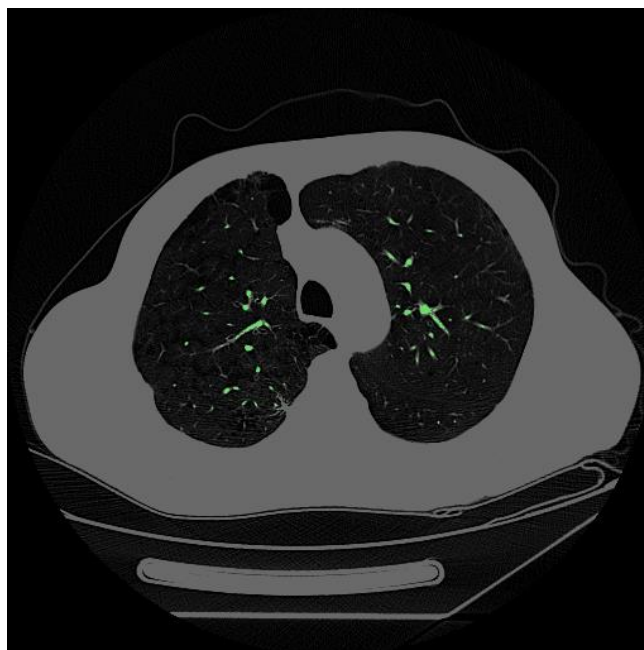


Рисунок 17. Результат работы модуля окраски

### 3 Проектирование веб-приложения для анализа снимков компьютерной томографии на предмет наличия областей кальцинации

#### 3.1 Описание ролей пользователей

В веб-приложении предусмотрено три роли пользователей: *Врач*, *Администратор* и *Неавторизованный пользователь*. На рисунке 18 представлена диаграмма вариантов использования, отражающая все роли пользователей и прецеденты.

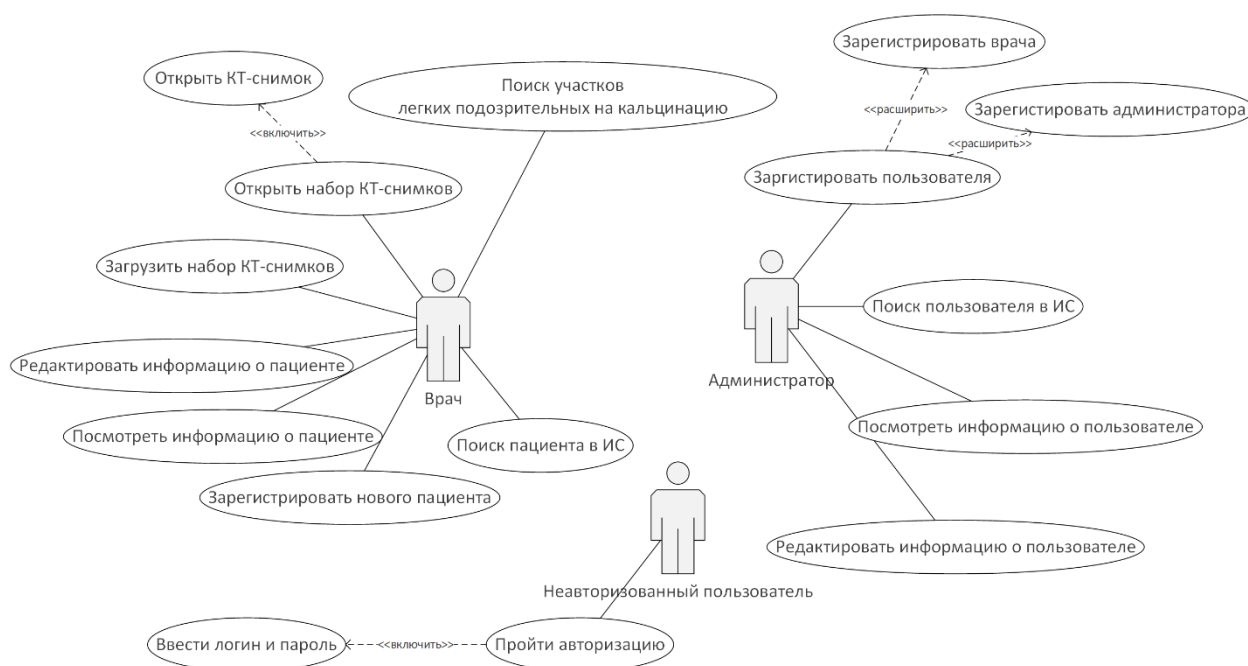


Рисунок 18. Диаграмма вариантов использования

#### 3.2 Проектирование веб-приложения с использованием методологий EPC и DFD

Для представления того, как веб-приложение будет внедрено в процесс обслуживания пациента при диагностике, была составлена событийная цепочка процессов – EPC-диаграмма (рисунок 19).

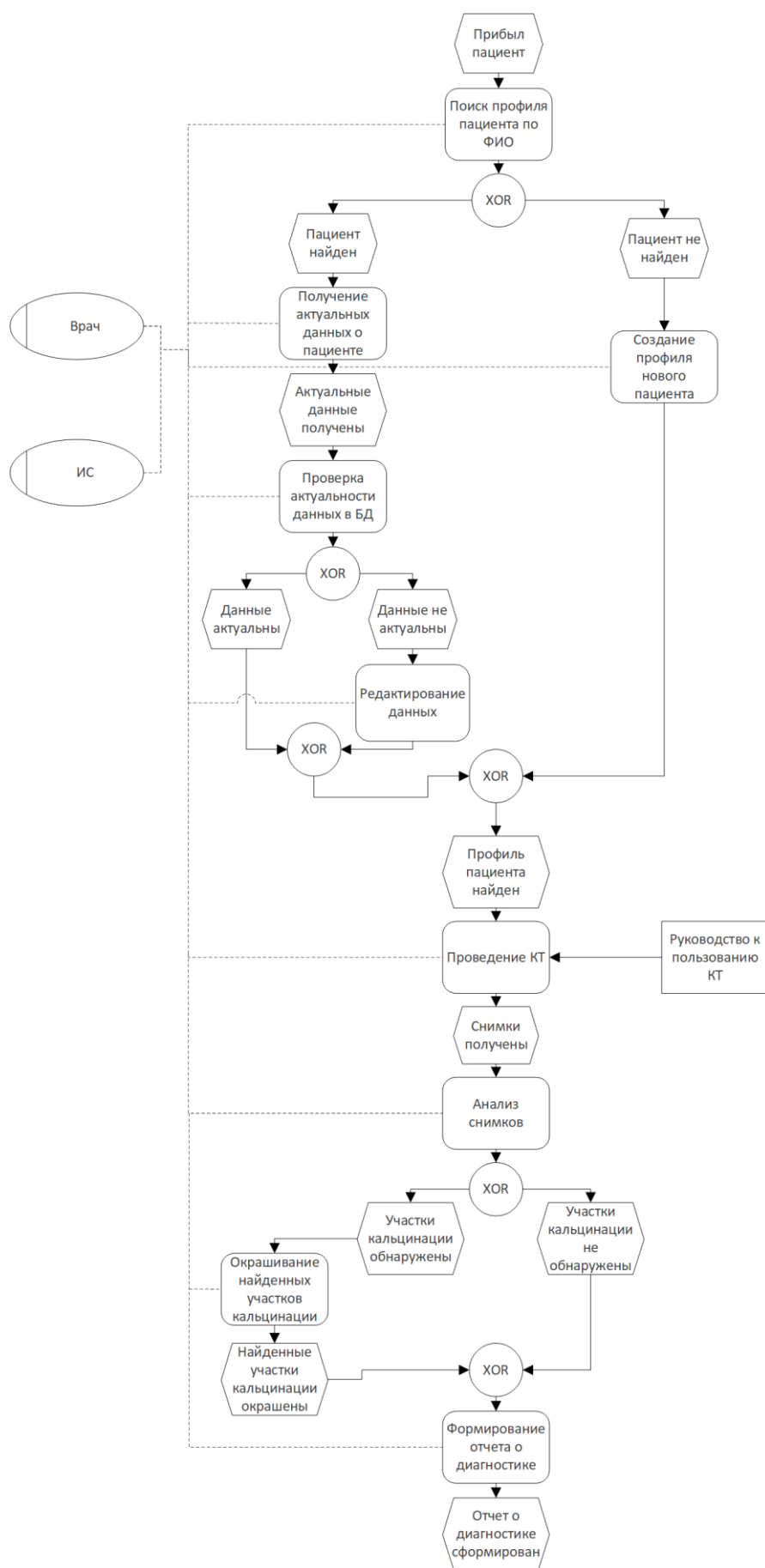


Рисунок 19. Событийная цепочка процессов



Кроме того, была составлена диаграмма потоков данных – DFD-диаграмма (рисунок 20).

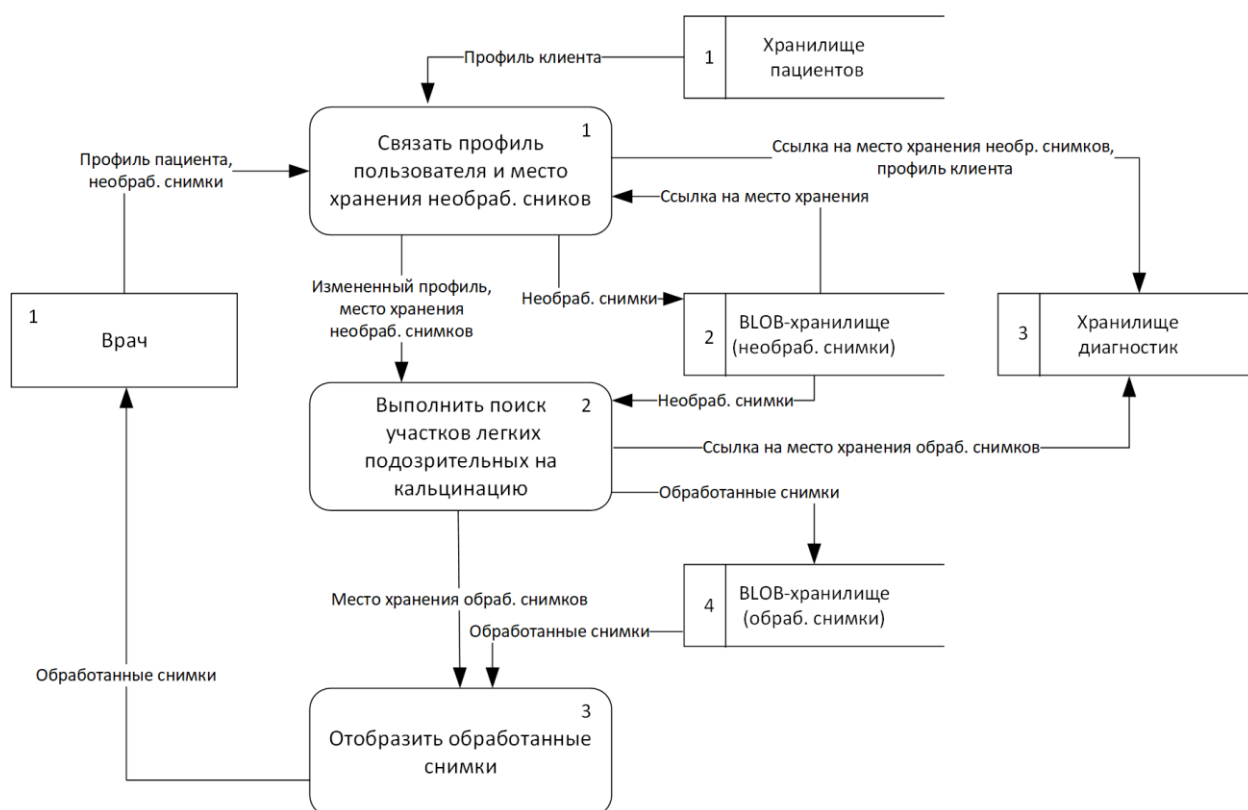


Рисунок 20. Диаграмма потоков данных

### 3.3 Проектирование базы данных веб-приложения с использованием методологии ERD

Для веб-приложения необходима база данных, в которой будут храниться данные о пользователях веб-приложения (администраторах и врачах), данные о пациентах и проведенных диагностиках, а также о месте хранения необработанных и обработанных снимков в BLOB-хранилище.

На рисунке 21 приведена логическая модель базы данных, построенная с помощью методологии ERD.

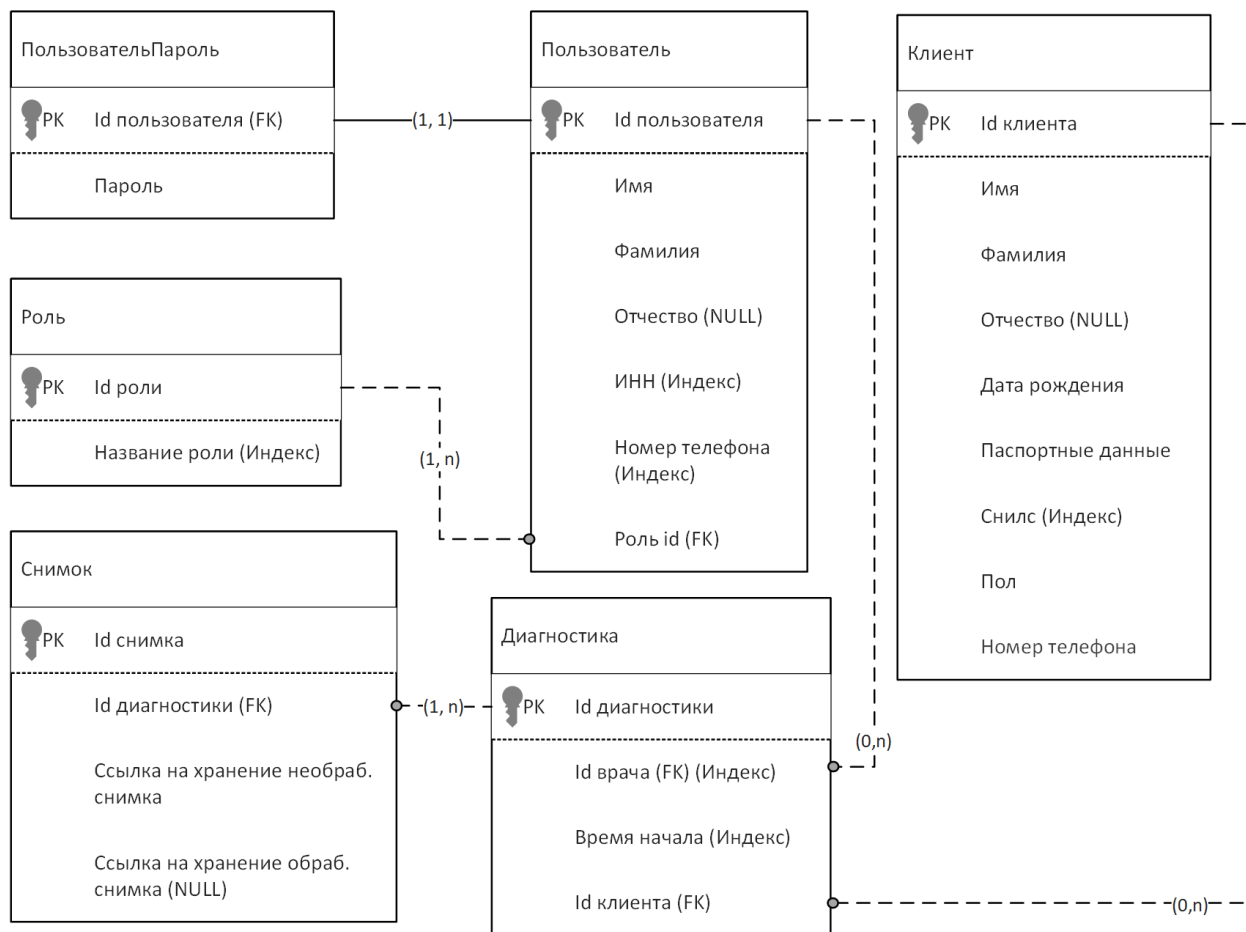


Рисунок 21. Логическая модель базы данных

База данных состоит из 6 таблиц.

Таблица «Пользователь» хранит данные о всех пользователях системы (администраторах и врачах). Роль пользователя определяется таблицей «Роль». Пароли пользователей хранятся в отдельной таблице «ПользовательПароль».

Таблица «Клиент» хранит данные о всех пациентах поликлиники.

Данные о всех диагностиках записываются в таблицу «Диагностика». Данные о снимках, полученных в ходе диагностики, сохраняются в таблице «Снимок». Сами снимки хранятся в облачном хранилище, в таблице хранятся только ссылки на место в хранилище.

### 3.4 Проектирование диаграммы классов анализа веб-приложения

Для формирования представления о структуре создаваемого проекта веб-приложения была создана диаграмма классов анализа (рисунок 22).

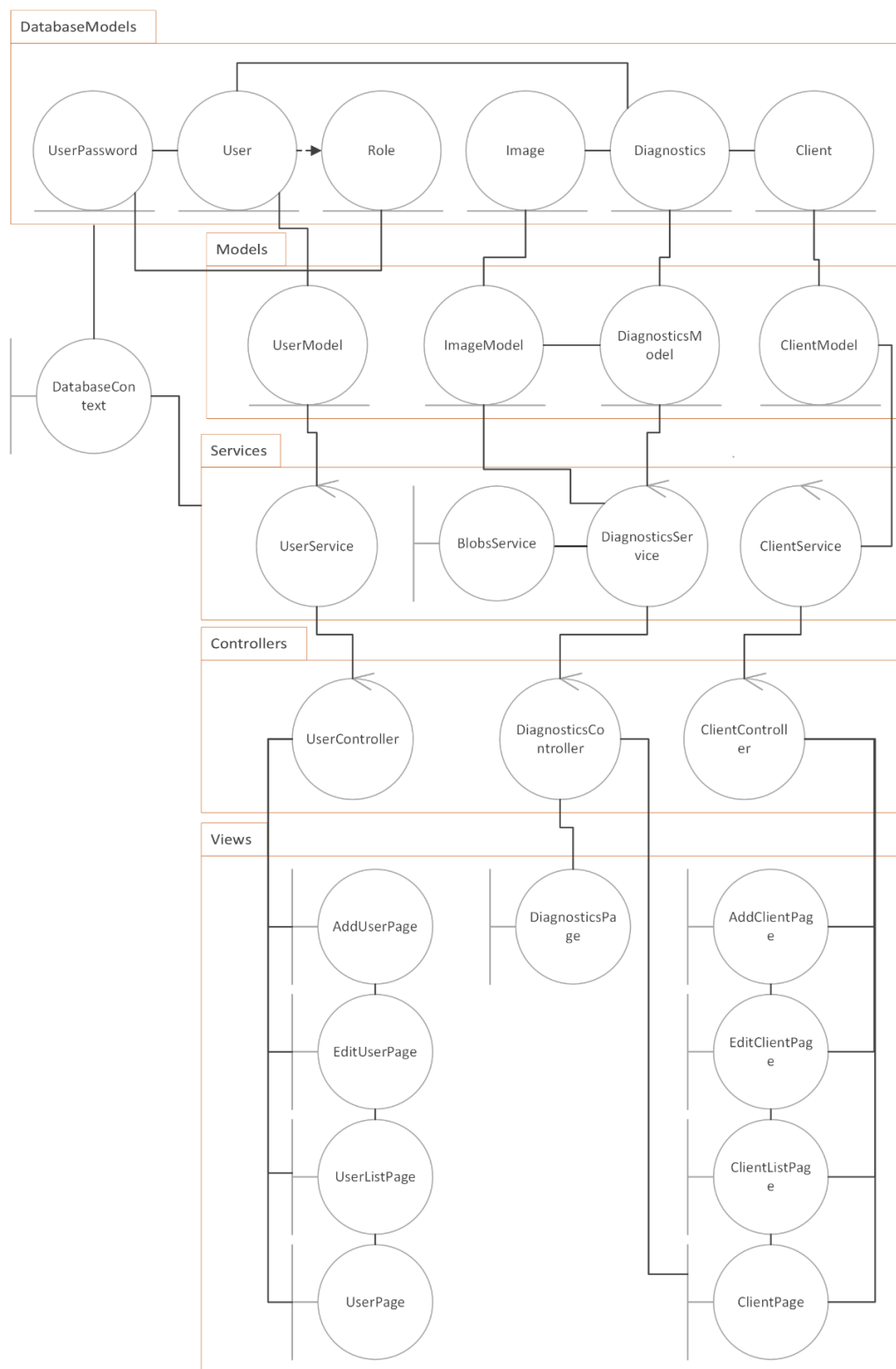


Рисунок 22. Диаграмма классов анализа

Пояснения к диаграмме классов анализа приведены в таблице Б.1 приложения Б.

### 3.5 Проектирование макетов страниц веб-приложения

Проектирование интерфейса производилось с помощью онлайн-сервиса «NinjaMock» [10]. В этом сервисе были созданы макеты будущих страниц веб-приложения (рисунки 23-25).

Список пациентов

Вы вошли как Фамилия И.О. [Выйти](#)

Поиск

Поиск по

☐ СНИЛС

☒ Фамилии

Фамилия	Имя	Отчество	Пол	Дата рождения	Номер паспорта	Номер СНИЛС	Телефон	
								<a href="#">Профиль</a> <a href="#">Диагностика</a>
								<a href="#">Профиль</a> <a href="#">Диагностика</a>
								<a href="#">Профиль</a> <a href="#">Диагностика</a>

[Добавить пациента](#)

<< 1 2 3 4 5 >>

Рисунок 23. Макет страницы «Список пациентов»



Добавить пациента

Вы вошли как **Фамилия И.О.** [Выйти](#)

Информация о пациенте

Фамилия

Фамилия

Имя

Имя

Отчество

Отчество

СНИЛС

СНИЛС

Пол

Пол

✓

Дата рождения

Дата рождения

Номер паспорта

Номер паспорта

Номер телефона

Номер телефона

Отмена

Сохранить

Рисунок 25. Макет страницы «Добавить пациента»

## **4 Разработка веб-приложения для анализа снимков компьютерной томографии на предмет наличия областей кальцинации**

### **4.1 Обоснование выбора средств разработки веб-приложения**

Перед началом разработки веб-приложения необходимо было определиться, какой будет тип приложения (многостраничный – ASP.Net Core MVC, или одностраничный - ASP.Net Core Web API), какой фреймворк использовать для создания клиентской части приложения в случае одностраничного веб-приложения (Angular или Vue.js), а также какую платформу выбрать для облачного хранилища (Microsoft Azure, Amazon Web Services или Google Cloud Platform). Для этого были составлены матрицы морфологического анализа для возможных вариантов (таблица 2, 3).

Таблица 2 – Выбор типа веб-приложения

<b>Критерий</b>	<b>Весовой коэффициент критерия</b>	<b>Вариант</b>		
		ASP.Net Core MVC	ASP.Net Core Web API + Angular	ASP.Net Core Web API + Vue.js
Опыт работы	0,2	4	4	4
Кроссплатформенность	0,4	5	5	5
Документация и поддержка	0,2	5	5	4
Скорость работы	0,2	4	5	5
ИТОГО	1	4,6	4,8	4,6

Таблица 3 – Выбор платформы для облачного хранилища [11]

Критерий	Весовой коэффициент критерия	Вариант		
		Microsoft Azure	Amazon Web Services	Google Cloud Platform
Опыт работы	0,2	3	0	0
Наличие возможности резервного копирования данных	0,1	5	2	0
Документация и поддержка	0,3	5	5	5
Длительность бесплатной пробной версии	0,5	4	5	5
ИТОГО	1	4,6	4,2	4

Тип приложения – ASP.Net Core Web API + Angular. Такое приложение будет кроссплатформенным, так как для запуска пользователю нужен будет только интернет-браузер. Кроме того, такое приложение будет одностраничным (SPA), что позволит увеличить скорость работы (по сравнению с ASP.Net Core MVC), так как не будет тратиться время на полную перезагрузку страницы при каждом запросе. Документация и поддержка для Angular имеет более высокий уровень по сравнению с Vue.js по причине большей распространенности фреймворка Angular.



Платформой для облачного хранилища выбрана платформа Microsoft Azure. С этой платформой имеется опыт работы, поэтому не придется затрачивать время на изучение способа взаимодействия с хранилищем.

#### **4.2 Взаимодействие веб-приложения с модулем обнаружения областей кальцинации**

Веб-приложение и модуль обнаружения областей кальцинации написаны на разных языках программирования (C# и Python соответственно). По этой причине было невозможно включить модуль в веб-приложение напрямую, а было решено создать два разных приложения, которые будут взаимодействовать друг с другом.

Взаимодействие между приложением обнаружения областей кальцинации и веб-приложением осуществляется через сокеты, где приложение Python является сервером, а веб-приложение – клиентом. В листингах В.1 и В.2 приложения В приведены функции, осуществляющие подключение и общение между этими приложениями.

#### **4.3 Результат разработки веб-приложения**

Результатом разработки веб-приложения являются серверная и клиентская часть приложения. В приложении Г для примера содержания этих частей приведены листинги для поиска клиентов по фамилии или СНИЛС (листинги Г.1-Г.4).

Основными страницами разработанного веб-приложения являются страницы «Список пациентов» и «Диагностика».

На рисунке 26 представлена страница «Список пациентов».

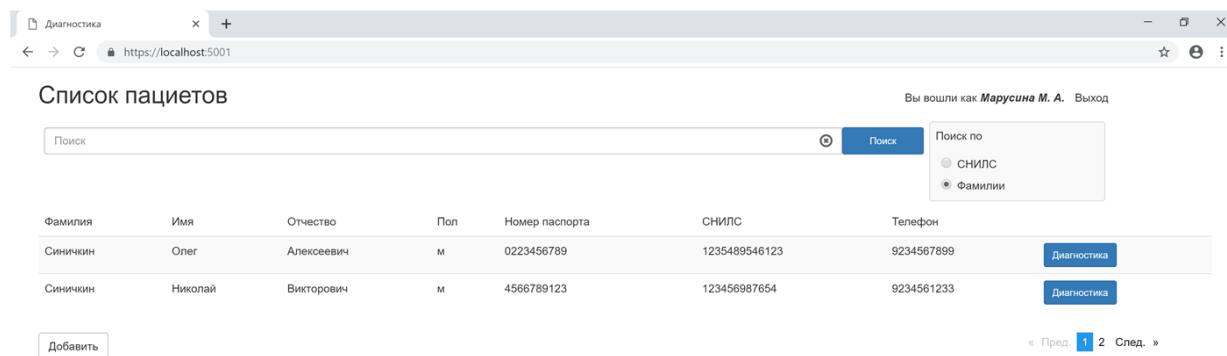


Рисунок 26. Страница «Список пациентов»

На странице «Список пациентов» пользователь может выполнить поиск пациентов по фамилии или номеру СНИЛС (предварительно выбрав необходимый параметр поиска). Если пациента в базе данных еще нет, его можно создать, нажав на кнопку «Добавить» и введя данные пациента. После того, как необходимый профиль пациента найден, пользователю необходимо нажать кнопку «Диагностика» напротив данных выбранного пациента. После этого пользователь попадает на страницу «Диагностика» (рисунок 27).

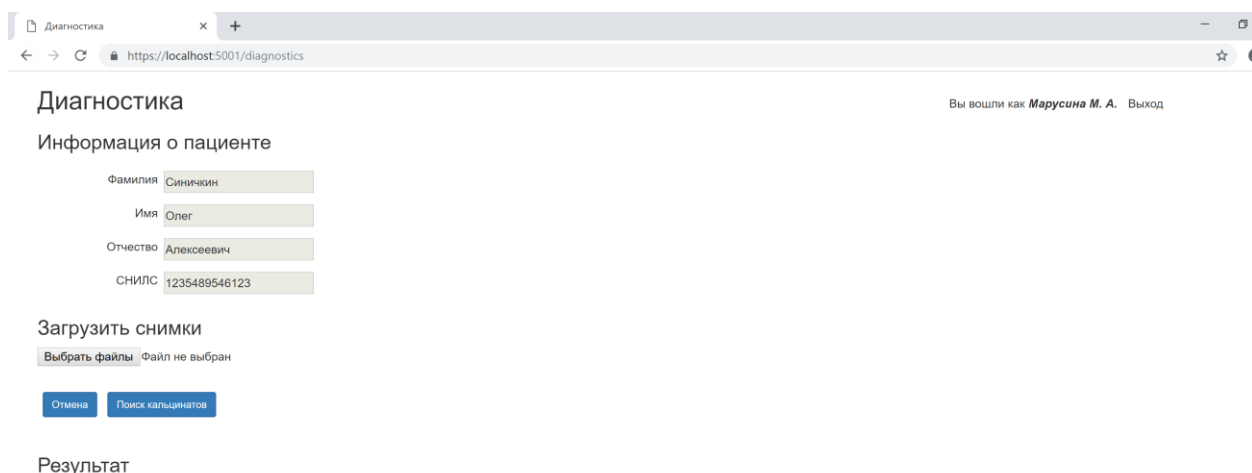


Рисунок 27. Страница «Диагностика» до обработки снимков

При переходе на страницу «Диагностика», пользователь должен выбрать КТ-снимки для анализа, нажав кнопку «Выбрать файлы». После этого в разделе «Результат» появятся обработанные снимки (рисунок 28).

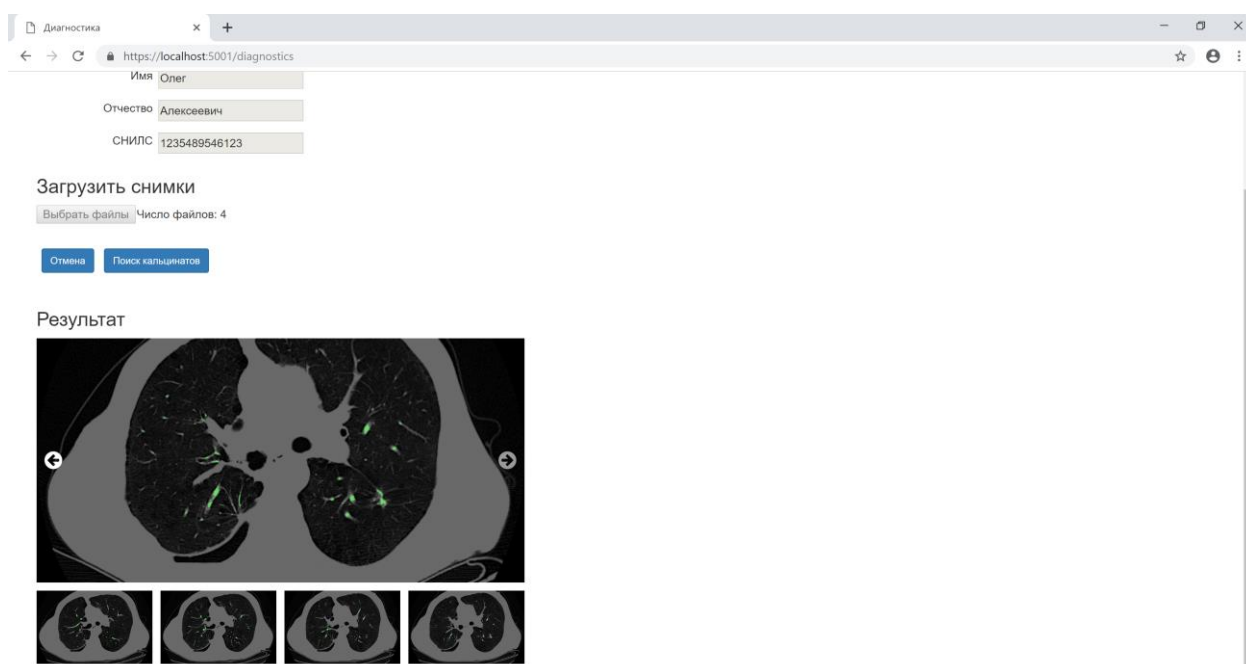


Рисунок 28. Страница «Диагностика» после обработки снимков

## **5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение**

### **Введение**

В настоящее время во многие области медицины проникают информационные технологии. Автоматизированные системы позволяют более точно и быстро устанавливать наличие того или иного заболевания.

Одной из медицинских задач, информационное решение для которой еще не разработано, является автоматизированный поиск участков кальциноза в легочной системе по снимкам компьютерной томографии. Кальциноз – это заболевание, при котором в тканях появляются кальцинаты – образования, формируемые в результате скопления солей кальция.

В ходе данной работы было создано веб-приложение, которое после загрузки пользователем КТ-снимков анализирует их, выполняя поиск подозрительных на кальциноз участков, окрашивает предполагаемые участки и выдает уже обработанные снимки.

### **Цели**

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является проектирование и создание конкурентоспособных разработок, технологий, отвечающих современным требованиям в области ресурсоэффективности и ресурсосбережения.

В рамках выбранной тематики проведения научного исследования «Локализация областей кальцинации в лёгочной системе по снимками компьютерной томографии» целесообразным является:

1. Определить востребованность разработки;
2. Определить преимущества и недостатки продукта;
3. Оценить возможные варианты продвижения разработки на рынке;

4. Осуществить детальное планирование разработки приложения;
5. Выбрать наиболее эффективный способ реализации приложения.

### **Задачи**

Для достижения описанных выше целей необходимо решить следующие задачи:

1. Определение потенциальных потребителей путем рассмотрения целевого рынка и проведения его сегментирования;
2. QuaD-анализ для корректировки собственного проекта;
3. Проведение SWOT-анализа для комплексной оценки угроз, возможностей, сильных и слабых сторон научно-исследовательского проекта, а также для анализа внешней и внутренней среды проекта;
4. Расчёт затрат на реализацию проекта;
5. Расчёт эффективности различных вариантов реализации приложения.

## **5.1 Оценка коммерческого потенциала**

### **5.1.1 Потенциальные потребители результатов исследования**

Проведем анализ рынка потенциальных потребителей. Потенциальными потребителями выступают медицинские учреждения, которые практикуют диагностику заболеваний лёгочной системы и которым важно, чтобы процесс постановки диагноза был незамедлительным и быстрым.

Разработка предполагает её использование совместно с компьютерным томографом. Соответственно, в местах, где применяется КТ для анализа легочной системы, разрабатываемая система сможет быть применена.

### 5.1.2 QuaD-анализ

Из-за специфики разрабатываемого алгоритма (поиск участков кальцинации в лёгочной системе) полных аналогов найдено не было. Для оценки качества и перспективности разрабатываемой системы был проведен анализ по технологии QuaD (таблица 4).

Таблица 4 – Анализ по технологии QuaD

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение	Средневзвешенное значение (*100)
1	2	3	4	5	6
<b>Показатели оценки качества разработки</b>					
1. Потребность в ресурсах памяти	0,05	70	100	0,7	3,5
2. Функциональность	0,1	90	100	0,9	9
3. Простота эксплуатации	0,1	80	100	0,8	8
4. Скорость работы	0,15	95	100	0,95	14,25
5. Надежность	0,15	70	100	0,7	10,5
6. Удобство эксплуатации	0,1	85	100	0,85	8,5
7. Точность	0,2	80	100	0,8	16
<b>Показатели оценки коммерческого потенциала разработки</b>					
8. Конкурентоспособность продукта	0,03	80	100	0,8	2,4
9. Перспективность рынка	0,05	60	100	0,6	3
10.Срок выхода на рынок	0,05	50	100	0,5	2,5

#### Окончание таблицы 4

11.Послепродажное сопровождение	0,02	75	100	0,75	1,5
<b>Итого</b>	<b>1</b>	<b>835</b>	<b>1100</b>	<b>8,35</b>	<b>79,15</b>

Средневзвешенное значение показателя качества и перспективности разрабатываемой системы равна 79,15. Таким образом, перспективность данной разработки выше среднего (находится в пределах от 60-79).

#### 5.1.3 SWOT-анализ

SWOT – Strengths (сильные стороны), Weaknesses (слабые стороны), Opportunities (возможности) и Threats (угрозы) – представляет собой комплексный анализ научно-исследовательского проекта. SWOT-анализ применяют для исследования внешней и внутренней среды проекта. Сильные и слабые стороны являются факторами внутренней среды объекта анализа, возможности и угрозы являются факторами внешней среды.

Результат проведения SWOT-анализа приведен в таблице 5.

Таблица 5 – Матрица SWOT

	<p><b>Сильные стороны научно-исследовательского проекта:</b></p> <p>С1. Использование алгоритмов машинного обучения.</p> <p>С2. Уникальность разработки.</p> <p>С3. Направленность на решение проблем, связанных со здоровьем.</p> <p>С4. Возможность добавления поиска других заболеваний легочной системы.</p>	<p><b>Слабые стороны научно-исследовательского проекта:</b></p> <p>Сл1. Большие временные затраты на подготовку тестовой выборки.</p> <p>Сл2. Узкая направленность разработки.</p>
--	--	--

## Окончание таблицы 5

<b>Возможности:</b> В1. Появление новых алгоритмов машинного обучения. В2. Наличие отзывов от потребителей. В3. Появление новых потребителей.	Необходимо следить за отзывами потребителей и подстраиваться под их нужды, добавляя возможность поиска требуемых им заболеваний.	Необходимо следить за появлением новых алгоритмов, которые позволят сократить временные затраты.
<b>Угрозы:</b> У1. Появление или увеличение конкуренции. У2. Трудность распространения информации о разработке до потенциальных заказчиков. У3. Отсутствие спроса на внедрение разработки.	Направленность на решение проблем, связанных со здоровьем, привлечет заинтересованные стороны и позволит увеличить спрос на внедрение разработки.	Необходимо расширять направление использования разработки, то есть увеличивать количество заболеваний, которые можно определить по снимкам, что позволит повысить спрос на использование разработки.

Также была построена интерактивная матрица проекта для сильных сторон (таблица 6) и для слабых сторон (таблица 7).

Таблица 6 – Интерактивная матрица проекта (сильные стороны)

Сильные стороны проекта					
Возможности проекта		C1	C2	C3	C4
	B1	+	-	-	+
	B2	-	0	0	+
	B3	-	+	+	+



Таблица 7 – Интерактивная матрица проекта (слабые стороны)

Слабые стороны проекта			
Угрозы проекта		Сл1	Сл2
	У1	+	+
	У2	-	+
	У3	-	+

Таким образом, одним из основных направлений разработки является расширение количества заболеваний, поиск которых можно выполнять при анализе снимков.

## 5.2 Определение возможных альтернатив проведения научных исследований

Для создания стратегии разработки и продвижения проекта необходимо рассмотреть все возможные альтернативы создания продукта и выбрать из них наиболее оптимальные по тому или иному критерию. Это можно сделать, используя морфологический подход (таблица 8).

Таблица 8 – Морфологическая матрица проекта

	1	2	3
А. Язык реализации алгоритма	Python	C#	Java
Б. Платформа приложения	Веб-приложение ASP.Net Core MVC	Веб-приложение ASP.Net Core Web API + Angular	Десктопное приложение WPF
В. Платформа облачного хранилища	Microsoft Azure	Amazon Web Services	Google Cloud Platform
Г. Язык приложения	Русский	Английский	

В качестве наиболее вероятных для реализации вариантов были выбраны следующие:

- A1B2B1Г1
- A2B3B1Г1
- A1B3B3Г1

### 5.3 Планирование научно-исследовательских работ

#### 5.3.1 Структура работ в рамках научного исследования

Работу над проектом можно представить в виде следующих этапов (таблица 9).

Таблица 9 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб.	Содержание работ	Должность исполнителя
Этап подготовки			
Анализ конкурентных решений	1	Изучение существующих на рынке подобных решений	Руководитель, дипломник
Определение функциональности приложения	2	Выявление требований	Руководитель, дипломник
	3	Составление технического задания	Руководитель, дипломник
Определение сроков выполнения проекта	4	Календарное планирование работ	Руководитель, дипломник

Продолжение таблицы 9

Этап проектирования			
Проектирование архитектуры приложения	5	Построение архитектурной модели приложения	Дипломник
	6	Проектирование базы данных приложения	Дипломник
Определение платформы приложения	7	Определение операционной системы, для которой будет создаваться приложение	Руководитель, дипломник
Определение инструментальных средств разработки	8	Определение среды разработки для всех модулей приложения	Руководитель, дипломник
Изучение средств разработки	9	Изучение документации по выбранным средствам разработки	Дипломник
Проектирование интерфейса приложения	10	Создание макетов страниц приложения	Дипломник
Разработка алгоритма поиска областей кальцинации	11	Разработка эффективного алгоритма нахождения областей, подозрительных на кальцинацию, по КТ-снимкам легочной системы	Дипломник
Этап реализации			
Разработка приложения	12	Реализация алгоритма поиска областей кальцинации	Дипломник
	13	Создание и подключение базы данных	Дипломник

## Окончание таблицы 9

	14	Реализация логики приложения	Дипломник
	15	Объединение модулей приложения	Дипломник
Тестирование приложения	16	Выявление и устранение ошибок приложения	Дипломник
Развертывание приложения	17	Развертывание приложения в облачном хостинге	Дипломник

### 5.3.2 Определение трудоёмкости выполнения работ

Трудоёмкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов.

Для расчета длительности работ в календарных днях использовался коэффициент календарности равный 1,48, что соответствует 2019 календарному году.

Далее представлено рассчитываемое время для каждого их этапов проекта (таблица 10).

Таблица 10 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ									Исполни- тели	Длитель- ность работ в рабочих днях $T_{pi}$	Длитель- ность работ в календарных днях $T_{ki}$						
	$t_{min}$ , чел-дни			$t_{max}$ , чел-дни			$t_{ож\kern 0.05em i}$ , чел-дни											
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3				Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Анализ конкурентных решений	1	1	1	1	1	1	1	1	1	Р, Д	Р, Д	Р, Д	0,5	0,5	0,5	0,7 4	0,7 4	0,7 4

# Окончание таблицы 10

Определение функциональности приложения	1	1	1	5	5	5	3	3	3	Р, Д	Р, Д	Р, Д	1,5	1,5	1,5	2,22	2,22	2,22
Определение сроков выполнения проекта	1	1	1	1	1	1	1	1	1	Р, Д	Р, Д	Р, Д	0,5	0,5	0,5	0,74	0,74	0,74
Проектирование архитектуры приложения	1	1	1	5	5	5	3	3	3	Д	Д	Д	3	3	3	4,43	4,43	4,43
Определение платформы приложения	1	1	1	1	1	1	1	1	1	Р, Д	Р, Д	Р, Д	0,5	0,5	0,5	0,74	0,74	0,74
Определение инструментальных средств разработки	1	1	1	1	1	1	1	1	1	Р, Д	Р, Д	Р, Д	0,5	0,5	0,5	0,74	0,74	0,74
Изучение средств разработки	5	5	5	10	10	15	7	7	12	Д	Д	Д	7	7	12	10,34	10,34	17,73
Проектирование интерфейса приложения	2	2	2	2	2	2	2	2	2	Д	Д	Д	2	2	2	2,96	2,96	2,96
Разработка алгоритма поиска областей кальцинации	10	10	10	20	25	20	18	22	18	Д	Д	Д	18	22	18	26,6	32,51	26,6
Разработка приложения	25	25	35	40	40	50	35	35	45	Д	Д	Д	35	35	45	51,72	51,72	66,5
Тестирование приложения	5	5	5	10	10	10	8	8	8	Д	Д	Д	8	8	8	11,82	11,82	11,82
Развертывание приложения	1	1	1	2	2	2	1	1	1	Д	Д	Д	1	1	1	1,48	1,48	1,48
<b>Итого:</b>	Всего												77,5	81,5	92,5	115	121	137
	Руководитель												3,5	3,5	3,5	5,18	5,18	5,18
	Дипломник												77,5	81,5	92,5	115	121	137

На основе временных показателей был построен календарный план-график научного исследовательского проекта. График строится для

максимального по длительности исполнения работ в рамках научно-исследовательского проекта с разбивкой по месяцам и неделям за период дипломирования. Для удобной визуализации разработка календарного плана графика производится в программе Microsoft Project 2016 в виде диаграммы Ганта (рисунок 29).

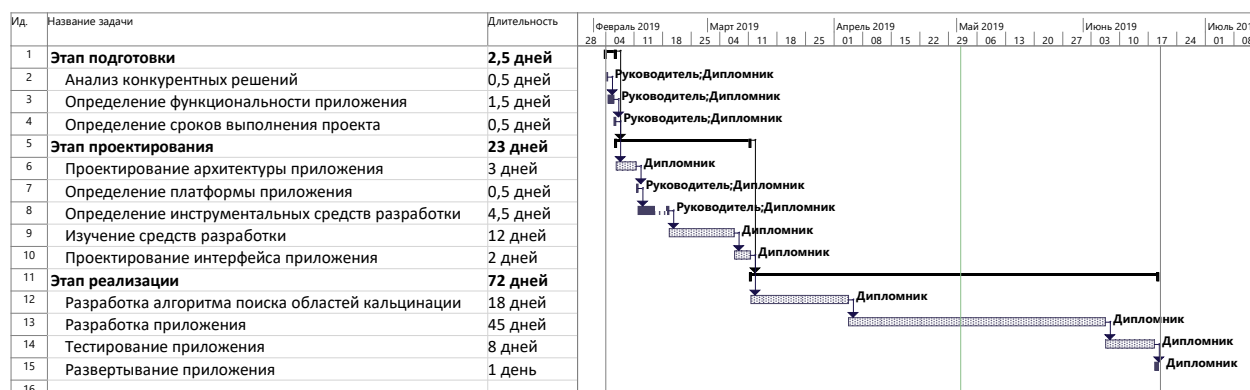


Рисунок 29. Календарный план-график работ

### 5.3.3 Бюджет научно-технического исследования (НТИ)

При планировании бюджета НТИ должно быть обеспечено полное и достоверное представление всех видов расходов, связанных с его выполнением. В процессе формирования бюджета НТИ были рассмотрены следующие статьи:

1. основная заработная плата исполнителей темы;
2. дополнительная заработная плата исполнителей темы;
3. отчисления во внебюджетные фонды (страховые отчисления);
4. накладные расходы.

Материальных затрат НТИ, затрат на специальное оборудование для научных (экспериментальных) работ, затрат на научные и производственные командировки, контрагентных расходов в ходе выполнения проекта не предвидится.

### 5.3.3.1 Основная заработная плата исполнителей темы

Планирование основной заработной платы производилось из расчета того, что основная заработная плата руководителя составляет 1500 руб/день, дипломника – 900 руб/день.

Планирование основной заработной платы приведено в таблице 11.

Таблица 11 – Расчет основной заработной платы

Название работы	Трудоёмкость работы			Заработная плата, приходящаяся на один чел.-день, руб.			Всего заработная плата по тарифу (окладам)		
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Анализ конкурентных решений	0,5	0,5	0,5	1500+900	1500+900	1500+900	1200	1200	1200
Определение функциональности приложения	1,5	1,5	1,5	1500+900	1500+900	1500+900	3600	3600	3600
Определение сроков выполнения проекта	0,5	0,5	0,5	1500+900	1500+900	1500+900	1200	1200	1200
Проектирование архитектуры приложения	3	3	3	900	900	900	2700	2700	2700
Определение платформы приложения	0,5	0,5	0,5	1500+900	1500+900	1500+900	1200	1200	1200
Определение инструментальных средств разработки	0,5	0,5	0,5	1500+900	1500+900	1500+900	1200	1200	1200
Изучение средств разработки	7	7	12	900	900	900	6300	6300	10800
Проектирование интерфейса приложения	2	2	2	900	900	900	1800	1800	1800

### Окончание таблицы 11

Разработка алгоритма поиска областей кальцинации	18	22	18	900	900	900	16200	19800	16200
Разработка приложения	35	35	45	900	900	900	31500	31500	40500
Тестирование приложения	8	8	8	900	900	900	7200	7200	7200
Развертывание приложения	1	1	1	900	900	900	900	900	900
<b>Итого:</b>							75000	78600	88500

Таблица 12 – Баланс рабочего времени

<b>Показатели рабочего времени</b>	<b>Руководитель</b>	<b>Инженер</b>
Календарное число дней	365	365
Количество нерабочих дней – выходные дни – праздничные дни	118	118
Потери рабочего времени – отпуск – невыходы по болезни	30	30
Действительный годовой фонд рабочего времени	217	217



Расчет основной заработной платы исполнителям производился для максимального по длительности исполнения работ (таблица 13).

Таблица 13 – Расчет основной заработной платы исполнителям

Исполнитель	Разряд	$Z_{гс}$ , руб.	$k_{пр}$	$k_d$	$k_p$	$Z_m$ , руб	$Z_{дн}$ , руб	$T_p$ , раб. дн.	$Z_{осн}$ , руб
Руководитель	Ведущий программист	34615	-	-	1,3	45000	1500	3,5	5250
Дипломник	Младший программист	20769	-	-	1,3	27000	900	92,5	83250
Итого $Z_{осн}$									88500

### 5.3.3.2 Дополнительная заработная плата исполнителей темы

Расчет дополнительной заработной платы ведется по формуле 6:

$$Z_{доп} = k_{доп} * Z_{осн}, \quad (6)$$

где  $Z_{доп}$  – затраты по дополнительной заработной плате исполнителей темы;

$k_{доп}$  – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15);

$Z_{осн}$  – затраты по основной заработной плате исполнителей темы.

В данном случае коэффициент дополнительной заработной платы будет взят равным 0,13.

Таким образом, затраты на дополнительную заработную плату для максимального по длительности исполнения работ можно считать равными 11505 рублей.

### 5.3.3.3 Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из формулы 7:

$$З_{внеб} = k_{внеб} * (З_{осн} + З_{доп}), \quad (7)$$

где  $З_{внеб}$  – затраты на отчисления во внебюджетные фонды;

$k_{внеб}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.);

$З_{осн}$  – затраты по основной заработной плате исполнителей темы;

$З_{доп}$  – затраты по дополнительной заработной плате исполнителей темы.

Для учреждений, осуществляющих научную и образовательную деятельность, коэффициент отчислений берётся равным 28% (таблица 14).

Таблица 14 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб			Дополнительная заработная плата, руб		
	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Руководитель	5250	5250	5250	682,5	682,5	682,5

## Окончание таблицы 14

Дипломник	69750	73350	83250	9067,5	9535,5	10822,5
Коэффициент отчислений во внебюджетные фонды	0,28					
Итого						
Исп. 1	23714,88					
Исп. 2	24869,04					
Исп. 3	28001,4					

**5.3.3.4 Накладные расходы**

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле 8:

$$Z_{\text{накл}} = (\text{сумма статей } 1 \div 3) * k_{\text{нр}}, \quad (8)$$

где  $Z_{\text{накл}}$  – затраты на накладные расходы;

$k_{\text{нр}}$  – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов можно взять в размере 16%.

Таким образом, накладные расходы для максимального по длительности исполнения работ можно считать равными 20481,02 рублей.

### 5.3.3.5 Формирование бюджета затрат научно-исследовательского проекта

По итогам всех рассчитанных статей для каждого из вариантов исполнения приведён расчёт бюджета затрат НТИ (таблица 15).

Таблица 15 – Расчет бюджета затрат НТИ

Наименование статьи	Сумма, руб			Примечание
	Исп. 1	Исп. 2	Исп. 3	
1. Затраты по основной заработной плате исполнителей темы	75000	78600	88500	Пункт 1.3.3.1
2. Затраты по дополнительной заработной плате исполнителей темы	9750	10218	11505	Пункт 1.3.3.2
3. Отчисления во внебюджетные фонды	23714,88	24869,04	28001,4	Пункт 1.3.3.3
4. Накладные расходы	17354,38	18189,93	20481,02	16% от суммы ст. 1-3
5. Бюджет затрат НТИ	125819,26	131876,97	148487,42	Сумма ст. 1-4

#### 5.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальное и экономической эффективности исследования

Определение эффективности проекта осуществляется с помощью расчета интегрального показателя эффективности научного исследования. Для этого необходимо определить две средневзвешенные величины: финансовую эффективность и ресурсоэффективность.

Расчёт финансовой эффективности проводится с помощью расчёта интегрального финансового показателя, который вычисляется по формуле 9:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (9)$$

где  $I_{\text{финр}}^{\text{исп.}i}$  – интегральный финансовый показатель разработки;

$\Phi_{pi}$  – стоимость  $i$ -го варианта исполнения;

$\Phi_{\text{max}}$  – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Таблица 16 – Расчет интегрального финансового показателя

	Исп. 1	Исп. 2	Исп. 3
Бюджет затрат	125819,26	131876,97	148487,42
Интегральный финансовый показатель	0,85	0,89	1

Интегральный показатель ресурсоэффективности рассчитывается как сумма балльных оценок каждого из вариантов в матрице сравнительной оценки по критериям с учётом их весовых коэффициентов (таблица 17).

Таблица 17 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Весовой коэффициент параметра	Объект исследования		
		Исп. 1	Исп. 2	Исп. 3
Скорость работы	0,15	5	3	4
Точность	0,15	5	4	5
Потребность в ресурсах памяти	0,1	4	5	4
Функциональность	0,2	5	5	5
Кроссплатформенность	0,15	5	2	5
Наличие возможности резервного копирования данных	0,05	4	4	2
Удобство эксплуатации	0,1	5	4	4
Простота эксплуатации	0,1	5	4	5
ИТОГО	1	4,85	3,85	4,5

Интегральный показатель эффективности определяется по формуле 10:

$$I_{\text{исп.}i} = \frac{I_{p-\text{исп.}i}}{I_{\text{финр}}^{\text{исп.}i}}, \quad (10)$$

где  $I_{\text{исп.}i}$  – интегральный показатель эффективности  $i$ -го исполнения;

$I_{p-\text{исп.}i}$  – интегральный показатель ресурсоэффективности  $i$ -го исполнения;

$I_{\text{финр}}^{\text{исп.}i}$  – интегральный финансовый показатель  $i$ -го исполнения.

В таблице 18 рассчитан интегральный показатель эффективности для каждого из вариантов исполнений, а также приведена их сравнительная эффективность.

Таблица 18 – Сравнительная эффективность разработки

№ п/п	Показатель	Исп. 1	Исп. 2	Исп. 3
1	Интегральный финансовый показатель разработки	0,85	0,89	1
2	Интегральный показатель ресурсоэффективности разработки	4,85	3,85	4,5
3	Интегральный показатель эффективности	5,7	4,32	4,5
4	Сравнительная эффективность вариантов исполнения	1,27	0,96	1

### Заключение

В результате работы по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» были выявлены и сравнены три варианта исполнения научно-исследовательской работы. Наиболее эффективным вариантом исполнения оказался вариант исполнения 1 – А1Б2В1Г1 (язык реализации алгоритма – Python, платформа приложения – ASP.Net Core Web API + Angular (веб-приложение), платформа облачного хранилища – Microsoft Azure, язык приложения – русский). Бюджет затрат этого варианта составляет 125819,26 рублей, а срок выполнения – 115 календарных дней.

## **6 Социальная ответственность**

### **Введение**

В настоящее время во многие области медицины проникают информационные технологии. Автоматизированные системы позволяют более точно и быстро устанавливать наличие того или иного заболевания.

Одной из медицинских задач, информационное решение для которой еще не разработано, является автоматизированный поиск участков кальциноза в легочной системе по снимкам компьютерной томографии. Кальциноз – это заболевание, при котором в тканях появляются кальцинаты – образования, формируемые в результате скопления солей кальция.

В ходе данной работы было создано веб-приложение, которое после загрузки пользователем КТ-снимков анализирует их, выполняя поиск подозрительных на кальциноз участков, окрашивает предполагаемые участки и выдает уже обработанные снимки.

Научно-исследовательский проект представляет собой разработку программного продукта и предполагает большой объем работы с ПК, поэтому важным критерием безопасности является организация рабочего места и режима трудовой деятельности. К вредным факторам труда инженера-программиста относятся: недостаточная освещенность рабочей зоны, отклонение параметров микроклимата в помещении и уровень шума, а к опасным факторам – опасность поражения электрическим током [12].

### **6.1 Правовые и организационные вопросы обеспечения безопасности**

#### **6.1.1 Специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства**

Проектируемая рабочая ситуация представляет собой непрерывную в течении всего рабочего дня работу за ПК. Правовыми нормами трудового законодательства для этого варианта рабочей зоны являются:



1. ТК РФ;
2. типовая инструкция ТОИ Р-45-084-01;
3. СанПиН 2.2.2. / 2.4. 1340-03.

Согласно типовой инструкции ТОИ Р-45-084-01 [13] особенностью трудового законодательства для рассматриваемой области является разделение видов трудовой деятельности за ПК на три группы:

1. А – работа по считыванию информации с экрана компьютера с предварительным запросом;
2. Б – работа по вводу информации;
3. В – творческая работа в режиме диалога с компьютером.

Разработку ПО можно относить к группе В. Разделение категорий работ по уровню нагрузки распределяется в этой группе представлено в таблице 19.

Таблица 19 – Распределение уровня нагрузки по категориями работ для группы В

<b>Категория работы</b>	<b>Уровень нагрузки за рабочую смену, час.</b>
I	до 2,0
II	до 4,0
III	до 6,0

То есть, в данном случае имеем категорию работ III группы В. В этом случае необходимо устраивать перерывы через 1,5 - 2,0 часа от начала рабочей смены и через 1,5 - 2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут

через каждый час работы для того, чтобы исключить повышенную утомляемость и снижение работоспособности.

### **6.1.2 Организационные мероприятия при компоновке рабочей зоны**

В типовой инструкции ТОИ Р-45-084-01 [13] для рабочей зоны инженера-программиста устанавливаются следующие правила:

1. рабочие места с компьютерами должны размещаться таким образом, чтобы расстояние от экрана одного видеомонитора до тыла другого было не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м;

2. рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, преимущественно слева;

3. оконные проемы в помещениях, где используются персональные компьютеры, должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.;

4. рабочая мебель для пользователей компьютерной техникой должна отвечать следующим требованиям:

4.1. высота рабочей поверхности стола должна регулироваться в пределах 680 - 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм;

4.2. рабочий стол должен иметь пространство для ног высотой не менее 600 мм, глубиной на уровне колен не менее 450 мм и на уровне вытянутых ног не менее 650 мм;

4.3. рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также по расстоянию спинки от переднего края сиденья;

4.4. рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов; поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

## 6.2 Профессиональная социальная безопасность

В таблице 20 представлены вредные и опасные факторы при разработке ПО.

Таблица 20 – Возможные вредные и опасные факторы

Факторы (ГОСТ 12.0.003-2015) [12]	Вид работ		Нормативные документы
	Разработка ПО	Применение ПО	
Вредные			
1. Отклонение показателей микроклимата	+	+	СанПиН 2.2.4.548–96 [14]
2. Превышение уровня шума	+	+	СН 2.2.4/2.1.8.562–96 [15], СНиП 23-03-2003 [16]
3. Недостаточная освещенность рабочей зоны	+	+	СП 52.13330.2016 [17], СанПиН 2.2.2/2.4.1340- 03 [18]
Опасные			
4. Опасность поражения электрическим током	+	+	СанПиН 2.2.2/2.4.1340- 03 [18] ГОСТ Р 50571.17-2000 [19]

## **6.2.1 Анализ вредных факторов, которые могут возникнуть при проведении исследований**

### **6.2.1.1 Отклонение показателей микроклимата**

Использование персональных компьютеров может привести к повышению температуры и снижению относительной влажности в рабочем помещении, то есть к изменению микроклимата. Отклонение показателей микроклимата от комфортных может повлиять на здоровье персонала.

Оптимальные параметры микроклимата определяются в зависимости от категории работ по уровню энергозатрат. Инженер-программист относится к категории Ia [20]. В таблице 21 приведены оптимальные и допустимые значения показателей микроклимата для рабочего помещения персонала категории Ia.

Таблица 21 – Оптимальные и допустимые величины показателей микроклимата на рабочих местах производственных помещений для категории Ia [14]

<b>Период года</b>	<b>Категория работ по уровню энергозатрат, Вт</b>	<b>Температура воздуха, °С</b>	<b>Температура поверхности, °С</b>	<b>Относительная влажность воздуха, %</b>	<b>Скорость движения воздуха, м/с</b>
<b>Оптимальные величины</b>					
Холодный	Ia (до 139)	22-24	21-25	60-40	0,1
Теплый	Ia (до 139)	23-25	22-26	60-40	0,1
<b>Допустимые величины</b>					
Холодный	Ia (до 139)	20-25	19-26	15-75	0,1
Теплый	Ia (до 139)	21-28	20-29	15-75	0,1-0,2

Для поддержания соответствующих микроклиматических параметров используются системы отопления и вентиляции, а также проводится кондиционирование воздуха в помещении.

#### **6.2.1.2 Превышение уровня шума**

Использование персональных компьютеров, наличие центральной системы вентиляции и кондиционирования воздуха приводят к повышению уровня шума на рабочем месте. Повышенный уровень шума затрудняет разборчивость речи, снижает работоспособность, повышает утомляемость.

В таблице 22 представлены предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для инженера-программиста.

Таблица 22 – Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для инженера-программиста [15]

<b>Вид трудовой деятельности, рабочее место</b>	<b>Уровни звукового давления, дБ в октавных полосах со среднегеометрическими частотами, Гц</b>						
	31,5	63	125	250	500	1000	2000
Творческая деятельность, руководящая работа с повышенными требованиями, научная деятельность, конструирование и проектирование, программирование, преподавание и обучение, врачебная деятельность. Рабочие места в помещениях дирекции, проектно-конструкторских бюро, расчетчиков, программистов вычислительных машин, в лабораториях для теоретических работ и обработки данных, приема больных в здравпунктах	86	71	61	54	49	45	42

Для уменьшения воздействий шума можно использовать следующие методы [16]:

1. экранирование рабочих мест, то есть установка перегородок между рабочими местами;
2. установка оборудования, производящего минимальный шум.

Системы охлаждения являются основной причиной шума персональных компьютеров. Существуют разные системы охлаждения с уровнем шума до 60 дБ. Тихой системой охлаждения считается та, у которой уровень шума в пределах 20-25 дБ [21]. Этим можно руководствоваться при выборе оборудования для рабочего места.

Тем не менее, любое оборудование при загрязнении увеличивает уровень шума, поэтому рекомендуется регулярно проводить его техническое обслуживание: чистка от пыли, замена смазывающих веществ.

#### **6.2.1.3 Недостаточная освещенность рабочей зоны**

Отсутствие достаточного количества источников освещения в рабочей зоне является причиной такого вредного фактора, как недостаточная освещенность. Недостаточная освещенность приводит к понижению работоспособности, а также может вызвать проблемы со здоровьем, а именно может повлиять на качество зрения персонала.

Согласно СП 52.13330.2016 [17] зрительную работу инженера-программиста можно характеризовать как работу разряда Б – высокой точности (наименьший эквивалентный размер объекта различения составляет 0,3-0,5 мм), подразряда 1 (относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%). В таблице 23 представлены требования к освещению рабочего помещения для указанного разряда.

Таблица 23 – Требования к освещению рабочего помещения для разряда Б1 [17]

Искусственное освещение				Естественное освещение	
Освещенность на рабочей поверхности от системы общего освещения, лк	Цилиндрическая освещенность, лк	Объединенный показатель дискомфорта, не более	Коэффициент пульсации освещенности, $K_n$ , %, не более	Коэффициент естественной освещенности, %, при	
				верхнем или комбинированном	боковом
300	100	21	15	3	1

Для снижения влияния фактора недостаточной освещенности необходимо, чтобы уровень естественного освещения рабочего места и яркость дисплея персонального компьютера были приблизительно одинаковыми, так как яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение, что приводит к их быстрой утомляемости.

Также следует соблюдать следующие нормативы, установленные в СанПиН 2.2.2/2.4.1340-03 [18]:

1. следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране персонального компьютера не должна превышать  $40 \text{ кд/м}^2$  и яркость потолка не должна превышать  $200 \text{ кд/м}^2$ ;

2. дизайн персонального компьютера должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света; корпус персонального компьютера, клавиатура и другие блоки и устройства персонального компьютера должны иметь матовую поверхность с коэффициентом отражения 0,4-0,6 и не иметь блестящих деталей, способных создавать блики;

3. для освещения помещений с персонального компьютера следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пускорегулирующими аппаратами; допускается использование многоламповых светильников с электромагнитными пускорегулирующими аппаратами, состоящими из равного числа опережающих и отстающих ветвей.

## **6.2.2 Анализ опасных факторов, которые могут возникнуть при проведении исследований**

### **6.2.2.1 Опасность поражения электрическим током**

Электрические установки, к которым относится персональный компьютер, представляют для человека большую потенциальную опасность, так как в процессе эксплуатации или проведения профилактических работ человек может коснуться комплектующих компьютера, находящихся под напряжением. Кроме того, корпуса персонального компьютера и прочего оборудования, оказавшегося под напряжением в результате повреждения или пробоя изоляции, не подают каких-либо сигналов, которые предупреждают человека об опасности. Причинами электропоражений являются: провода с поврежденной изоляцией, розетки сети без предохранительных кожухов. Поражение персонала электрическим током является опасным для жизни и здоровья.

Согласно СанПиН 2.2.2/2.4.1340-03 [18] рабочее место инженера-программиста должно находиться в безопасной зоне, которое не характеризуется наличием таких условий, как повышенная влажность (относительная влажность воздуха длительно превышает 75%), высокая температура (более 35°C), токопроводящая пыль, токопроводящие полы, возможность одновременного соприкосновения к имеющим соединения с землей металлическим элементам и металлическим корпусам электрооборудования.



Для защиты от поражения электрическим током все токоведущие части должны быть защищены от случайных прикосновений кожухами, корпус устройства должен быть заземлен. Заземление выполняется изолированным медным проводом сечением 1,5 мм, который присоединяется к общей шине заземления с общим сечением 48 мм при помощи сварки. Общая шина присоединяется к заземлению, сопротивление которого не должно превышать 4 Ом.

Согласно ГОСТ Р 50571.17-2000 [19] питание устройства в рабочем помещении инженера-программиста должно осуществляться от силового щита через автоматический предохранитель, который срабатывает при коротком замыкании нагрузки. Для снижения величин возникающих разрядов должны применяться покрытия из антистатического материала. Рабочее место инженера-программиста должно быть оборудовано таким образом, чтобы исключить взаимное соприкосновение кабелей и шнуров питания соседних компьютеров.

К организационно-техническим мероприятиям относится первичный инструктаж по технике безопасности, который является обязательным условием для допуска к работе в данном помещении.

### **6.3 Экологическая безопасность**

Разрабатываемое программное обеспечение не оказывает влияния на окружающую среду, так как разрабатывается и используется внутри персональных компьютеров.

Тем не менее, некоторые средства исследования могут стать источником загрязнения окружающей среды, а именно:

1. персональные компьютеры;
2. люминесцентные лампы.

### **6.3.1 Анализ воздействия средств исследования на окружающую среду**

Для уменьшения вредного воздействия от утилизации ПК и люминесцентных ламп на окружающую среду необходимо соблюдать правила утилизации.

Согласно «Методике проведения работ по комплексной утилизации вторичных драгоценных металлов из отработанных средств вычислительной техники» [22] утилизация персональных компьютеров предусматривает 4 этапа, в рамках которых производится разборка и подготовка к утилизации внутренних частей ПК.

В результате выполнения первых 3-х этапов формируется партия сырья, включающая ячейки и элементы для объединений микросхем и электрорадиоэлементов, содержащие драгоценные металлы, а также партии черных и цветных металлов и сплавов (медь, сталь, никель, латунь, бронза, алюминий, дюралюминий, свинцово-оловянные припой), направляемых на переработку.

На последнем этапе осуществляется разборка ячеек и элементов для объединений микросхем и электрорадиоэлементов до уровня отдельных компонентов.

Согласно ГОСТ 12.3.031-83 [23] все ртутьсодержащие отходы и вышедшие из строя приборы, содержащие ртуть, подлежат сбору и возврату для последующей регенерации ртути в специализированных организациях. К работе по замене и сбору отработанных ртутьсодержащих ламп допускаются только электромонтеры. Главным условием при замене и сборе отработанных ртутьсодержащих ламп является сохранение герметичности. Факт сдачи ртутьсодержащих отходов подтверждается возвращением паспорта на вывоз отходов с отметкой о приеме представителя специализированного предприятия.

## **6.4 Безопасность в чрезвычайных ситуациях**

Наиболее вероятными ЧС для рассматриваемой рабочей зоны являются:

1. авария на коммунальных системах жизнеобеспечения;
2. пожар (взрыв) в здании;
3. землетрясение.

### **6.4.1 Обоснование мероприятий по предотвращению наиболее типичной ЧС и разработка порядка действий в случае возникновения ЧС**

Источниками ЧС – пожар (взрыв) в здании, может стать электропроводка, внутренние работающие устройства ПК, взрывоопасные предметы в рабочем помещении.

Превентивными мерами по предупреждению ЧС могут служить системы звукового и визуального оповещения персонала лаборатории и кабинетов об опасности, обучение персонала, методам работы с компьютером, наличие средств пожаротушения и информационных досок с планами эвакуации.

Согласно «Постановлению о противопожарном режиме» [24] порядок действий при возникновении пожара для каждого гражданина следующий:

1. незамедлительно сообщить об этом по телефону в пожарную охрану (при этом необходимо назвать адрес объекта, место возникновения пожара, а также сообщить свою фамилию);
2. принять по возможности меры по эвакуации людей, тушению пожара и сохранности материальных ценностей.

Также, согласно «Постановлению о противопожарном режиме» [24], лица, уполномоченные владеть, пользоваться или распоряжаться имуществом, руководители и должностные лица организаций, лица, в

установленном порядке назначенные ответственными за обеспечение пожарной безопасности, по прибытии к месту пожара должны:

1. сообщить о возникновении пожара в пожарную охрану, поставить в известность руководство и дежурные службы объекта;
2. в случае угрозы жизни людей немедленно организовать их спасание, используя для этого имеющиеся силы и средства;
3. проверить включение в работу автоматических систем противопожарной защиты (оповещения людей о пожаре, пожаротушения, противодымной защиты);
4. при необходимости отключить электроэнергию (за исключением систем противопожарной защиты), остановить работу транспортирующих устройств, агрегатов, аппаратов, перекрыть сырьевые, газовые, паровые и водяные коммуникации, остановить работу систем вентиляции в аварийном и смежном с ним помещениях, выполнить другие мероприятия, способствующие предотвращению развития пожара и задымления помещений здания;
5. прекратить все работы в здании (если это допустимо по технологическому процессу производства), кроме работ, связанных с мероприятиями по ликвидации пожара;
6. удалить за пределы опасной зоны всех работников, не участвующих в тушении пожара;
7. осуществить общее руководство по тушению пожара (с учетом специфических особенностей объекта) до прибытия подразделения пожарной охраны;
8. обеспечить соблюдение требований безопасности работниками, принимающими участие в тушении пожара;

9. одновременно с тушением пожара организовать эвакуацию и защиту материальных ценностей;

10. организовать встречу подразделений пожарной охраны и оказать помощь в выборе кратчайшего пути для подъезда к очагу пожара;

11. сообщать подразделениям пожарной охраны, привлекаемым для тушения пожаров и проведения связанных с ними первоочередных аварийно-спасательных работ, сведения о перерабатываемых или хранящихся на объекте опасных (взрывоопасных), взрывчатых, сильнодействующих ядовитых веществах, необходимые для обеспечения безопасности личного состава.

12. по прибытии пожарного подразделения информировать руководителя тушения пожара о конструктивных и технологических особенностях объекта защиты, прилегающих строений и сооружений, о количестве и пожароопасных свойствах хранимых и применяемых на объекте защиты веществ, материалов, изделий и сообщение других сведений, необходимых для успешной ликвидации пожара.

### **Заключение**

В результате работы по разделу «Социальная ответственность» были выявлены основные нормативные акты для обеспечения безопасности жизнедеятельности на рабочем месте, рассмотрены наиболее значимые опасные и вредные факторы работы инженера-программиста, описано влияние разработки на окружающую среду, сформирован план действий при возникновении ЧС – пожар. Следуя правилам, описанным в данном разделе, можно достичь комфортной и безопасной работы при разработке ПО инженером-программистом.

## **Заключение**

В результате выполнения работы был разработан алгоритм обнаружения областей кальцинации в лёгочной системе по снимкам компьютерной томографии, а также было разработано веб-приложение, позволяющее выполнять поиск таких областей по загруженным снимкам компьютерной томографии.

Разработанный алгоритм и веб-приложение позволят сократить время анализа снимков компьютерной томографии, что позволит ускорить процесс постановки диагноза пациенту.

Алгоритм обнаружения областей кальцинации был разработан с помощью языка программирования Python и инструментов машинного обучения – модель, предсказывающая область кальцинации, построена с помощью классификатора «Градиентный бустинг деревьев решений».

Серверная часть веб-приложения была разработана на платформе ASP.Net Core, а клиентская часть – с помощью фреймворка Angular. То есть, веб-приложение является одностраничным. Также веб-приложение взаимодействует с облачным хранилищем BLOB-объектов платформы Microsoft Azure. В этом облачном хранилище находятся исходные снимки компьютерной томографии лёгочной системы, а также обработанные алгоритмом снимки с подсвеченными областями, которые подозрительны на наличие кальцинатов.

## Список использованных источников

1. Кальцинаты в лёгких: причины образования, способы устранения [Электронный ресурс]. // Pulmono.ru – Всё о болезнях дыхательных путей. Режим доступа: <https://pulmono.ru/legkie/drugie4/kaltsinaty-v-lyogkih>, свободный (дата обращения 22.02.2019)
2. Методология IDEF0 [Электронный ресурс]. // ITteach.RU. Режим доступа: <https://itteach.ru/bpwin/metodologiya-idef0#toc4>, свободный (дата обращения 28.03.2019)
3. Искусственный интеллект в радиологии [Электронный ресурс]. // ZDRAV.EXPERT – Медтех-портал. Режим доступа: [http://zdrav.expert/index.php/Статья:Искусственный\\_интеллект\\_в\\_радиологии](http://zdrav.expert/index.php/Статья:Искусственный_интеллект_в_радиологии), свободный (дата обращения 07.04.2019)
4. Java is one of the most energy-efficient languages, Python among least energy efficient [Электронный ресурс]. // jaxenter. Режим доступа: <https://jaxenter.com/energy-efficient-programming-languages-137264.html>, свободный (дата обращения 31.03.2019)
5. Belarus tuberculosis portal [Электронный ресурс]. Режим доступа: <http://tuberculosis.by/>, свободный (дата обращения 22.01.2019)
6. Module: feature skimage [Электронный ресурс]. // scikit-image – image processing in python. Режим доступа: <https://scikit-image.org/docs/dev/api/skimage.feature.html>, свободный (дата обращения 07.04.2019)
7. Введение в машинное обучение. Руководство для специалистов по работе с данными / Андреас Мюллер, Сара Гвидо – Москва, 2016-2017. – 393 с.

8. Обход в ширину [Электронный ресурс]. // Университет ИТМО. Режим доступа: [http://neerc.ifmo.ru/wiki/index.php?title=Обход\\_в\\_ширину](http://neerc.ifmo.ru/wiki/index.php?title=Обход_в_ширину), свободный (дата обращения 22.02.2019)
9. Morphological Transformations [Электронный ресурс]. // OpenCV. Режим доступа: [https://docs.opencv.org/3.0-beta/doc/py\\_tutorials/py\\_imgproc/py\\_morphological\\_ops/py\\_morphological\\_ops.html](https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html), свободный (дата обращения 25.02.2019)
10. NinjaMock online wireframe and mockup tool [Электронный ресурс]. // NinjaMock. Режим доступа: <https://ninjamock.com>, свободный (дата обращения 07.04.2019)
11. Сравнение услуг облачных провайдеров: Microsoft Azure, AWS или Google Cloud [Электронный ресурс]. // LA.BY – Cloud Computing | Software | GameDev. Режим доступа: <http://la.by/blog/sravnenie-uslug-oblachnyh-provayderov-microsoft-azure-aws-ili-google-cloud>, свободный (дата обращения 31.03.2019)
12. ГОСТ 12.0.003-2015 ССБТ. Опасные и вредные производственные факторы. Классификация.
13. ТОО Р-45-084-01. Типовая инструкция по охране труда при работе на персональном компьютере" (утв. Приказом Минсвязи РФ от 02.07.2001 N 162).
14. СанПиН 2.2.4.548–96. Гигиенические требования к микроклимату производственных помещений.
15. СН 2.2.4/2.1.8.562–96. Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории застройки.
16. СНиП 23-03-2003. Строительные нормы и правила российской федерации. Защита от шума.



17. СП 52.13330.2016. Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95\*.

18. СанПиН 2.2.2/2.4.1340-03. "Гигиенические требования к персональным электронно-вычислительным машинам и организации работы".

19. ГОСТ Р 50571.17-2000 (МЭК 60364-4-482-82). Электроустановки зданий. Требования по обеспечению безопасности. Выбор мер защиты в зависимости от внешних условий. Защита от пожара.

20. ГОСТ 33875-2016. Система газоснабжения. Магистральная трубопроводная транспортировка газа. Безопасные для здоровья человека условия пребывания и пользования зданиями и сооружениями. Микроклимат. Технические требования. Приложение Б (рекомендуемое). Категории работ по уровню общих энергозатрат работающего человека.

21. Словарь терминов. Кулеры и системы охлаждения компьютеров. [Электронный ресурс]. // Яндекс Маркет. Режим доступа: <https://market.yandex.ru/faq?hid=818965>, свободный (дата обращения 18.05.2019).

22. Методика проведения работ по комплексной утилизации вторичных драгоценных металлов из отработанных средств вычислительной техник, утвержденная Государственным Комитетом РФ по телекоммуникациям от 19 октября 1999 г.

23. ГОСТ 12.3.031-83. Система стандартов безопасности труда (ССБТ). Работы со ртутью. Требования безопасности.

24. Постановление правительства Российской Федерации «О противопожарном режиме» от 25 апреля 2012 года N 390 (с изменениями на 7 марта 2019 года).

## Приложение А

В листинге А.1 приведен код для построения графика сравнения точности рассмотренных классификаторов (k ближайших соседей, дерево решений, случайный лес, градиентный бустинг деревьев решений).

### Листинг А.1. Сравнение классификаторов

```
# Создаем массивы для хранения значений, по которым будут строиться
графики
training_accuracy = []
test_accuracy = []
test_test_accuracy = []
range_graf = []

# Получаем точность классификатора k ближайших соседей
knn = KNeighborsClassifier(n_neighbors=9)
knn.fit(X_train, y_train)
print("knn{:.3f}".format(knn.score(test_task_data, test_task_target)))
training_accuracy.append(knn.score(X_train, y_train))
test_accuracy.append(knn.score(X_test, y_test))
test_test_accuracy.append(knn.score(test_task_data, test_task_target))
range_graf.append("knn")

# Получаем точность классификатора дерево решений
tree_clf = DecisionTreeClassifier(max_depth=10, random_state=0)
tree_clf.fit(X_train, y_train)
print("tree{:.3f}".format(tree_clf.score(test_task_data,
test_task_target)))
training_accuracy.append(tree_clf.score(X_train, y_train))
test_accuracy.append(tree_clf.score(X_test, y_test))
test_test_accuracy.append(tree_clf.score(test_task_data,
test_task_target))
range_graf.append("tree")

# Получаем точность классификатора случайный лес
forest = RandomForestClassifier(n_estimators=10, max_features=8,
random_state=2)
forest.fit(X_train, y_train)
print("forest{:.3f}".format(forest.score(test_task_data,
test_task_target)))
training_accuracy.append(forest.score(X_train, y_train))
test_accuracy.append(forest.score(X_test, y_test))
test_test_accuracy.append(forest.score(test_task_data,
test_task_target))
range_graf.append("forest")

# Получаем точность классификатора градиентный бустинг решений
gbrt = GradientBoostingClassifier(random_state=0, max_depth=4,
learning_rate = 0.8)
gbrt.fit(X_train, y_train)
```

```

        print("boosting{:.3f}".format(gbrt.score(test_task_data,
test_task_target)))
    training_accuracy.append(gbrt.score(X_train, y_train))
    test_accuracy.append(gbrt.score(X_test, y_test))
    test_test_accuracy.append(gbrt.score(test_task_data,
test_task_target))
    range_graf.append("boosting")

    # Строим и сохраняем график
    plt.plot(range_graf, training_accuracy, label="точность на обучающем
наборе")
    plt.plot(range_graf, test_test_accuracy, label="точность на тестовом
наборе")
    plt.ylabel("Точность")
    plt.xlabel("Метод")
    plt.legend(loc='lower right')
    plt.savefig("comparing.png")

```

## Приложение Б

В таблице Б.1 представлено описание диаграммы классов анализа.

Таблица Б.1 – Пояснения к диаграмме классов

Название	Назначение	Связь
<b>Пакет <i>Models</i></b>		
<i>ClientModel</i>	<ul style="list-style-type: none"> <li>• Содержит информацию о клиенте из таблицы <i>Client</i> базы данных</li> <li>• Содержит список всех диагностик клиента в виде списка <i>DiagnosticsModel</i></li> </ul>	<ul style="list-style-type: none"> <li>• Отражает таблицу <i>Client</i> базы данных</li> <li>• Использует <i>DiagnosticsModel</i></li> <li>• Используется в <i>ClientService</i></li> <li>• Используется в <i>ClientController</i></li> </ul>
<i>UserModel</i>	<ul style="list-style-type: none"> <li>• Содержит информацию о пользователе из таблицы <i>User</i> базы данных</li> <li>• Содержит информацию о пользователе из таблицы <i>UserPassword</i> базы данных</li> </ul>	<ul style="list-style-type: none"> <li>• Отражает таблицу <i>User</i> базы данных</li> <li>• Отражает таблицу <i>UserPassword</i> базы данных</li> <li>• Используется в <i>UserService</i></li> <li>• Используется в <i>UserController</i></li> </ul>
<i>DiagnosticsModel</i>	<ul style="list-style-type: none"> <li>• Содержит информацию о диагностике из таблицы <i>Diagnostics</i> базы данных</li> <li>• Содержит список снимков диагностики как список <i>ImageModel</i></li> </ul>	<ul style="list-style-type: none"> <li>• Отражает таблицу <i>Diagnostics</i> базы данных</li> <li>• Использует <i>ImageModel</i></li> <li>• Используется в <i>DiagnosticsService</i></li> <li>• Используется в <i>DiagnosticsController</i></li> </ul>
<i>ImageModel</i>	<ul style="list-style-type: none"> <li>• Содержит информацию о снимке из таблицы <i>Image</i> базы данных</li> </ul>	<ul style="list-style-type: none"> <li>• Отражает таблицу <i>Image</i> базы данных</li> <li>• Используется в <i>DiagnosticsModel</i></li> <li>• </li> </ul>

Продолжение таблицы Б.1

Пакет <i>Services</i>		
<i>DiagnosticsService</i>	<ul style="list-style-type: none"> <li>• Создание новой диагностики для клиента</li> <li>• Получение списка диагностик (их идентификаторов) клиента по идентификатору клиента</li> <li>• Получение информации о диагностике по идентификатору приема</li> <li>• Получение снимков (их идентификаторов) для диагностики по идентификатору диагностики</li> <li>• Получение информации о снимках по идентификатору снимка</li> </ul>	<ul style="list-style-type: none"> <li>• Через интерфейс находится в <i>DiagnosticsController</i></li> <li>• Использует <i>DiagnosticsModel</i></li> <li>• Использует <i>ImageModel</i></li> <li>• Вспользует <i>ClientModel</i></li> </ul>
<i>UserService</i>	<ul style="list-style-type: none"> <li>• Авторизация пользователя</li> <li>• Выход из своего аккаунта пользователя</li> <li>• Создание нового пользователя</li> <li>• Редактирование информации о пользователе по идентификатору пользователя</li> <li>• Получение списка всех пользователей</li> <li>• Поиск пользователя по ФИО</li> <li>• Получение информации о пользователе по идентификатору пользователя</li> </ul>	<ul style="list-style-type: none"> <li>• Через интерфейс находится в <i>UserController</i></li> <li>• Использует <i>UserModel</i></li> </ul>

Окончание таблицы Б.1

<i>ClientService</i>	<ul style="list-style-type: none"> <li>• Создание нового клиента</li> <li>• Редактирование информации о клиенте по идентификатору клиента</li> <li>• Получение списка всех клиентов</li> <li>• Поиск клиента по ФИО</li> <li>• Получение информации о клиенте по идентификатору клиента</li> </ul>	<ul style="list-style-type: none"> <li>• Через интерфейс находится в <i>ClientController</i></li> <li>• Использует <i>ClientModel</i></li> </ul>
<i>BlobsService</i>	<ul style="list-style-type: none"> <li>• Добавление снимка в BLOB-хранилище</li> <li>• Получение снимка из BLOB-хранилища</li> </ul>	<ul style="list-style-type: none"> <li>• Через интерфейс находится в <i>DiagnosticsService</i></li> </ul>
Пакет <i>Controllers</i>		
<i>UserController</i>	<ul style="list-style-type: none"> <li>• Для действий сервиса <i>UserService</i> связывает их с соответствующими запросами из пользовательских интерфейсов</li> </ul>	<ul style="list-style-type: none"> <li>• Использует интерфейс от <i>UserService</i></li> <li>• Используется на страницах <i>AddUserPage</i>, <i>EditUserPage</i>, <i>UserListPage</i>, <i>UserPage</i></li> </ul>
<i>ClientController</i>	<ul style="list-style-type: none"> <li>• Для действий сервиса <i>ClientService</i> связывает их с соответствующими запросами из пользовательских интерфейсов</li> </ul>	<ul style="list-style-type: none"> <li>• Использует интерфейс от <i>ClientService</i></li> <li>• Используется на страницах <i>AddClientPage</i>, <i>EditClientPage</i>, <i>ClientListPage</i>, <i>ClientPage</i></li> </ul>
<i>DiagnosticsController</i>	<ul style="list-style-type: none"> <li>• Для действий сервиса <i>DiagnosticsService</i> связывает их с соответствующими запросами из пользовательских интерфейсов</li> </ul>	<ul style="list-style-type: none"> <li>• Использует интерфейс от <i>DiagnosticsService</i></li> <li>• Используется на страницах <i>DiagnosticsPage</i>, <i>ClientPage</i></li> </ul>

## Приложение В

В листингах В.1 и В.2 приведено взаимодействие между приложением для обнаружения областей кальцинации и веб-приложением.

Листинг В.1. Создание сервера и выполнение запроса приложением для обнаружения областей кальцинации

```
# Создаем сервер
serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serversocket.bind(('localhost', 8080))
serversocket.listen(5)
print("Server is waiting for connections.")

while True:
    # Ожидаем подключение клиента
    clientConnection, clientAddress = serversocket.accept()
    print("Connected client : " , clientAddress)

    # Получаем сообщение от клиента
    data = clientConnection.recv(1024)
    # Считываем из сообщения путь исходных снимков и путь для
сохранения обработанных снимков
    path_orig = ""
    path_parsed = ""
    message = data.decode()
    st = 0
    for i in range(len(data)):
        if (message[i] == '\x00'):
            st = i
            break
        path_orig+=message[i]
    flag = False
    for i in range(st, len(data)):
        if(flag == False):
            if (message[i] != '\x00'):
                flag = True
        if(flag==True):
            if (message[i] == '\x00'):
                break
            path_parsed+=message[i]

    # Проводим поиск областей кальцинации
    analyze(path_orig, path_parsed)

    # Отправляем клиенту сообщение об успешном завершении анализа
    clientConnection.send(bytes("Successfully analyzed on
Server", 'UTF-8'))

    # Отключаем клиента
    clientConnection.close()
    print("Server is waiting for connections.")
```

```
# Делаем задержку, чтобы цикл не сильно загружал процессор
time.sleep(0.1)
```

## Листинг В.2. Подключение к серверу и отправка запроса веб-приложением

```
private void FindCalcinatesRegions(string origPath, string parsedPath)
{
    // Создаем клиента
    var socket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
    var server = Glossary.SocketAdress;
    var port = Glossary.SocketPort;

    // Подключаемся к серверу
    socket.Connect(server, port);
    // Формируем сообщение с путем исходных снимков и путем
для обработанных снимков
    var dataOrigPath = Encoding.UTF8.GetBytes(origPath);
    var dataParsedPath = Encoding.UTF8.GetBytes(parsedPath);
    var message = new byte[1024];
    for (int i = 0; i < 512; i++)
    {
        if (i < dataOrigPath.Length)
        {
            message[i] = dataOrigPath[i];
        }
        if (i < dataParsedPath.Length)
        {
            message[i + 512] = dataParsedPath[i];
        }
    }

    // Посылаем сообщение серверу
    socket.Send(message);

    // Получаем ответ от сервера
    var dataAns = new byte[1024];
    var answerLen = socket.Receive(dataAns);

    // Отключаем клиента
    socket.Close();
}
```



## Приложение Г

В листингах Г.1-Г.4 приведен код для поиска клиентов по фамилии или СНИЛС.

Листинг Г.1. Функции файла *client.component.ts* (клиент) для поиска клиентов по фамилии или СНИЛС

```
search() {
  if (this.searchWord !== "") {
    if (this.filterType === "snils") {
      let client: ClientModel = new ClientModel();
      client.snils = this.searchWord;
      this.filter(client);
    }
    if (this.filterType === "lastname") {
      let client: ClientModel = new ClientModel();
      client.lastName = this.searchWord;
      this.filter(client);
    }
  }
}

filter(client: ClientModel) {
  this.dataService.filterClients(client)
    .subscribe((data: ResponseJson) => {
      if (data.statusCode === 200) {
        this.clients = data.value;
      }
    });
}
```

Листинг Г.2 Содержимое файла *client.service.ts* (клиент) для выполнения операций над клиентами

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { ClientModel } from './client';

@Injectable()
export class DataService {
  private url = "/api/clients";
  constructor(private http: HttpClient) {
  }
  getClients() {
    return this.http.get(this.url);
  }
  filterClients(clientModel: ClientModel) {
    return this.http.post(this.url + '/' + 'filter', clientModel);
  }
}
```

Листинг Г.3. Метод файла *ClientController.cs* (сервер) для получения и обработки запроса от клиента на поиск клиентов по фамилии или СНИЛС

```
[HttpPost("filter")]
public IActionResult Filter([FromBody]ClientModel clientModel)
{
    JsonResult result;
    try
    {
        var clientModels = new List<ClientModel>();
        if (clientModel.Snils != null)
        {
            var clientModelFull =
clientService.FilterSnils(clientModel.Snils);
            clientModels.Add(clientModelFull);
        }
        else if (clientModel.LastName != null)
        {
            clientModels =
(List<ClientModel>)clientService.FilterLastname(clientModel.LastName);
        }
        else
        {
            throw new Exception("Необходимо указать СНИЛС или
фамилию");
        }
        result = Json(Ok(clientModels));
    }
    catch (Exception ex)
    {
        var errorModel = errorService.BuildError(ex);
        result = Json(BadRequest(errorModel));
    }
    return result;
}
```

Листинг Г.4. Содержимое файла *ClientService.cs* (сервер) для выполнения операций над клиентами

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using DiagnosticsApp.DatabaseModels;
using DiagnosticsApp.Models;

namespace DiagnosticsApp.Services.Clients
{
    public class ClientService : IClientService
    {
        private DiagnosticsDBContext diagnosticsDBContext;
```

```

        public ClientService(DiagnosticsDBContext
diagnosticsDBContext)
        {
            this.diagnosticsDBContext = diagnosticsDBContext;
        }

        public IEnumerable<ClientModel> FilterLastname(string
lastname)
        {
            if(lastname == null)
            {
                throw new Exception("Необходимо ввести фамилию");
            }

            lastname = lastname.Trim().ToLower();
            var clients = diagnosticsDBContext.Client.Where(o =>
o.LastName == lastname);
            if(clients == null)
            {
                throw new Exception("Пользователей с заданной фамилией
не найдено");
            }

            var clientModels = BuildModels(clients);

            return clientModels;
        }

        public ClientModel FilterSnils(string snils)
        {
            if(snils == null)
            {
                throw new Exception("Необходимо ввести номер СНИЛС");
            }

            var client = diagnosticsDBContext.Client.SingleOrDefault(o
=> o.Snils == snils);
            if(client == null)
            {
                throw new Exception("Пациента с заданным СНИЛС не
найдено");
            }

            var clientModel = BuildModel(client);

            return clientModel;
        }

        public IEnumerable<ClientModel> Get()
        {
            var clients = diagnosticsDBContext.Client.AsQueryable();
            var clientModels = BuildModels(clients);

            return clientModels;
        }

```

```

private                                     IEnumerable<ClientModel>
BuildModels(IQueryable<Client> clients)
{
    var clientModels = new List<ClientModel>();
    foreach (var client in clients)
    {
        var clientModel = BuildModel(client);
        clientModels.Add(clientModel);
    }

    return clientModels;
}

private ClientModel BuildModel(Client client)
{
    var clientModel = new ClientModel()
    {
        BirthDate = client.BirthDate,
        ClientId = client.ClientId,
        FatherName = client.FatherName,
        FirstName = client.FirstName,
        IsMale = client.IsMale,
        LastName = client.LastName,
        Passport = client.Passport,
        PhoneNumber = client.PhoneNumber,
        Snils = client.Snils
    };

    return clientModel;
}
}

```